

CARMAweb analysis

Delafondia

March 8, 2012

Contents

1 Preprocessing of two color microarrays	1
1.1 Reading the raw data	1
1.2 Background correction	3
1.3 Within array normalization	101
1.4 Between array normalization	150
1.5 Saving the normalized M and A values	201
2 Replicate handling	202
3 Determining differentially expressed genes using test statistics	204
3.1 Definition of the sample groups	204
3.2 Prefiltering of the data	205
3.3 Calculating the raw p values	207
3.4 Correcting the p values for multiple testing	208
4 Determining differentially expressed genes using test statistics	214
4.1 Definition of the sample groups	214
4.2 Prefiltering of the data	215
4.3 Calculating the raw p values	217
4.4 Correcting the p values for multiple testing	218
5 Determining differentially expressed genes using test statistics	224
5.1 Definition of the sample groups	224
5.2 Prefiltering of the data	225
5.3 Calculating the raw p values	227
5.4 Correcting the p values for multiple testing	228

6 Determining differentially expressed genes using test statistics	234
6.1 Definition of the sample groups	234
6.2 Prefiltering of the data	235
6.3 Calculating the raw p values	237
6.4 Correcting the p values for multiple testing	238

List of Figures

1.1	Histogram of the array 1 (slide01_056.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	4
1.2	Histogram of the array 2 (slide02_065.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	5
1.3	Histogram of the array 3 (slide03_058.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	6
1.4	Histogram of the array 4 (slide04_059.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	7
1.5	Histogram of the array 5 (slide05_060.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	8
1.6	Histogram of the array 6 (slide06_053.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	9
1.7	Histogram of the array 7 (slide07_054.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	10
1.8	Histogram of the array 8 (slide08_055.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	11

1.9 Histogram of the array 9 (slide17_095.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	12
1.10 Histogram of the array 10 (slide18_097.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	13
1.11 Histogram of the array 11 (slide19_098.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	14
1.12 Histogram of the array 12 (slide20_099.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	15
1.13 Histogram of the array 13 (slide21_045.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	16
1.14 Histogram of the array 14 (slide22_083.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	17
1.15 Histogram of the array 15 (slide23_298.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	18
1.16 Histogram of the array 16 (slide24_085.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	19
1.17 Histogram of the array 17 (slide33_100.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	20
1.18 Histogram of the array 18 (slide34_061.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	21
1.19 Histogram of the array 19 (slide35_062.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	22

1.20 Histogram of the array 20 (slide36_064.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	23
1.21 Histogram of the array 21 (slide37_063.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	24
1.22 Histogram of the array 22 (slide38_300.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	25
1.23 Histogram of the array 23 (slide39_072.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	26
1.24 Histogram of the array 24 (slide40_073.gpr). Raw data before background correction The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.	27
1.25 MA plot of array 1 (slide01_056.gpr). Raw data before background correction.	28
1.26 MA plot of array 2 (slide02_065.gpr). Raw data before background correction.	29
1.27 MA plot of array 3 (slide03_058.gpr). Raw data before background correction.	30
1.28 MA plot of array 4 (slide04_059.gpr). Raw data before background correction.	31
1.29 MA plot of array 5 (slide05_060.gpr). Raw data before background correction.	32
1.30 MA plot of array 6 (slide06_053.gpr). Raw data before background correction.	33
1.31 MA plot of array 7 (slide07_054.gpr). Raw data before background correction.	34
1.32 MA plot of array 8 (slide08_055.gpr). Raw data before background correction.	35
1.33 MA plot of array 9 (slide17_095.gpr). Raw data before background correction.	36
1.34 MA plot of array 10 (slide18_097.gpr). Raw data before background correction.	37
1.35 MA plot of array 11 (slide19_098.gpr). Raw data before background correction.	38
1.36 MA plot of array 12 (slide20_099.gpr). Raw data before background correction.	39
1.37 MA plot of array 13 (slide21_045.gpr). Raw data before background correction.	40
1.38 MA plot of array 14 (slide22_083.gpr). Raw data before background correction.	41
1.39 MA plot of array 15 (slide23_298.gpr). Raw data before background correction.	42
1.40 MA plot of array 16 (slide24_085.gpr). Raw data before background correction.	43
1.41 MA plot of array 17 (slide33_100.gpr). Raw data before background correction.	44
1.42 MA plot of array 18 (slide34_061.gpr). Raw data before background correction.	45
1.43 MA plot of array 19 (slide35_062.gpr). Raw data before background correction.	46
1.44 MA plot of array 20 (slide36_064.gpr). Raw data before background correction.	47

1.45 MA plot of array 21 (slide37_063.gpr). Raw data before background correction.	48
1.46 MA plot of array 22 (slide38_300.gpr). Raw data before background correction.	49
1.47 MA plot of array 23 (slide39_072.gpr). Raw data before background correction.	50
1.48 MA plot of array 24 (slide40_073.gpr). Raw data before background correction.	51
1.49 Boxplots. Raw data before background correction.	52
1.50 MA plot of array 1 (slide01_056.gpr). Raw data after background correction.	53
1.51 Histogram of the array 1 (slide01_056.gpr). Raw data after background correction.	54
1.52 MA plot of array 2 (slide02_065.gpr). Raw data after background correction.	55
1.53 Histogram of the array 2 (slide02_065.gpr). Raw data after background correction.	56
1.54 MA plot of array 3 (slide03_058.gpr). Raw data after background correction.	57
1.55 Histogram of the array 3 (slide03_058.gpr). Raw data after background correction.	58
1.56 MA plot of array 4 (slide04_059.gpr). Raw data after background correction.	59
1.57 Histogram of the array 4 (slide04_059.gpr). Raw data after background correction.	60
1.58 MA plot of array 5 (slide05_060.gpr). Raw data after background correction.	61
1.59 Histogram of the array 5 (slide05_060.gpr). Raw data after background correction.	62
1.60 MA plot of array 6 (slide06_053.gpr). Raw data after background correction.	63
1.61 Histogram of the array 6 (slide06_053.gpr). Raw data after background correction.	64
1.62 MA plot of array 7 (slide07_054.gpr). Raw data after background correction.	65
1.63 Histogram of the array 7 (slide07_054.gpr). Raw data after background correction.	66
1.64 MA plot of array 8 (slide08_055.gpr). Raw data after background correction.	67
1.65 Histogram of the array 8 (slide08_055.gpr). Raw data after background correction.	68
1.66 MA plot of array 9 (slide17_095.gpr). Raw data after background correction.	69
1.67 Histogram of the array 9 (slide17_095.gpr). Raw data after background correction.	70
1.68 MA plot of array 10 (slide18_097.gpr). Raw data after background correction.	71
1.69 Histogram of the array 10 (slide18_097.gpr). Raw data after background correction.	72
1.70 MA plot of array 11 (slide19_098.gpr). Raw data after background correction.	73
1.71 Histogram of the array 11 (slide19_098.gpr). Raw data after background correction.	74
1.72 MA plot of array 12 (slide20_099.gpr). Raw data after background correction.	75
1.73 Histogram of the array 12 (slide20_099.gpr). Raw data after background correction.	76
1.74 MA plot of array 13 (slide21_045.gpr). Raw data after background correction.	77
1.75 Histogram of the array 13 (slide21_045.gpr). Raw data after background correction.	78
1.76 MA plot of array 14 (slide22_083.gpr). Raw data after background correction.	79
1.77 Histogram of the array 14 (slide22_083.gpr). Raw data after background correction.	80
1.78 MA plot of array 15 (slide23_298.gpr). Raw data after background correction.	81
1.79 Histogram of the array 15 (slide23_298.gpr). Raw data after background correction.	82

1.80 MA plot of array 16 (slide24_085.gpr). Raw data after background correction.	83
1.81 Histogram of the array 16 (slide24_085.gpr). Raw data after background correction.	84
1.82 MA plot of array 17 (slide33_100.gpr). Raw data after background correction.	85
1.83 Histogram of the array 17 (slide33_100.gpr). Raw data after background correction.	86
1.84 MA plot of array 18 (slide34_061.gpr). Raw data after background correction.	87
1.85 Histogram of the array 18 (slide34_061.gpr). Raw data after background correction.	88
1.86 MA plot of array 19 (slide35_062.gpr). Raw data after background correction.	89
1.87 Histogram of the array 19 (slide35_062.gpr). Raw data after background correction.	90
1.88 MA plot of array 20 (slide36_064.gpr). Raw data after background correction.	91
1.89 Histogram of the array 20 (slide36_064.gpr). Raw data after background correction.	92
1.90 MA plot of array 21 (slide37_063.gpr). Raw data after background correction.	93
1.91 Histogram of the array 21 (slide37_063.gpr). Raw data after background correction.	94
1.92 MA plot of array 22 (slide38_300.gpr). Raw data after background correction.	95
1.93 Histogram of the array 22 (slide38_300.gpr). Raw data after background correction.	96
1.94 MA plot of array 23 (slide39_072.gpr). Raw data after background correction.	97
1.95 Histogram of the array 23 (slide39_072.gpr). Raw data after background correction.	98
1.96 MA plot of array 24 (slide40_073.gpr). Raw data after background correction.	99
1.97 Histogram of the array 24 (slide40_073.gpr). Raw data after background correction.	100
1.98 Boxplots. Raw data after background correction.	101
1.99 MA plot of array 1 (slide01_056.gpr). Within array normalized data.	102
1.100Histogram of the array 1 (slide01_056.gpr). Within array normalized data.	103
1.101MA plot of array 2 (slide02_065.gpr). Within array normalized data.	104
1.102Histogram of the array 2 (slide02_065.gpr). Within array normalized data.	105
1.103MA plot of array 3 (slide03_058.gpr). Within array normalized data.	106
1.104Histogram of the array 3 (slide03_058.gpr). Within array normalized data.	107
1.105MA plot of array 4 (slide04_059.gpr). Within array normalized data.	108
1.106Histogram of the array 4 (slide04_059.gpr). Within array normalized data.	109
1.107MA plot of array 5 (slide05_060.gpr). Within array normalized data.	110
1.108Histogram of the array 5 (slide05_060.gpr). Within array normalized data.	111
1.109MA plot of array 6 (slide06_053.gpr). Within array normalized data.	112
1.110Histogram of the array 6 (slide06_053.gpr). Within array normalized data.	113
1.111MA plot of array 7 (slide07_054.gpr). Within array normalized data.	114
1.112Histogram of the array 7 (slide07_054.gpr). Within array normalized data.	115
1.113MA plot of array 8 (slide08_055.gpr). Within array normalized data.	116
1.114Histogram of the array 8 (slide08_055.gpr). Within array normalized data.	117

1.115MA plot of array 9 (slide17_095.gpr). Within array normalized data.	118
1.116Histogram of the array 9 (slide17_095.gpr). Within array normalized data.	119
1.117MA plot of array 10 (slide18_097.gpr). Within array normalized data.	120
1.118Histogram of the array 10 (slide18_097.gpr). Within array normalized data.	121
1.119MA plot of array 11 (slide19_098.gpr). Within array normalized data.	122
1.120Histogram of the array 11 (slide19_098.gpr). Within array normalized data.	123
1.121MA plot of array 12 (slide20_099.gpr). Within array normalized data.	124
1.122Histogram of the array 12 (slide20_099.gpr). Within array normalized data.	125
1.123MA plot of array 13 (slide21_045.gpr). Within array normalized data.	126
1.124Histogram of the array 13 (slide21_045.gpr). Within array normalized data.	127
1.125MA plot of array 14 (slide22_083.gpr). Within array normalized data.	128
1.126Histogram of the array 14 (slide22_083.gpr). Within array normalized data.	129
1.127MA plot of array 15 (slide23_298.gpr). Within array normalized data.	130
1.128Histogram of the array 15 (slide23_298.gpr). Within array normalized data.	131
1.129MA plot of array 16 (slide24_085.gpr). Within array normalized data.	132
1.130Histogram of the array 16 (slide24_085.gpr). Within array normalized data.	133
1.131MA plot of array 17 (slide33_100.gpr). Within array normalized data.	134
1.132Histogram of the array 17 (slide33_100.gpr). Within array normalized data.	135
1.133MA plot of array 18 (slide34_061.gpr). Within array normalized data.	136
1.134Histogram of the array 18 (slide34_061.gpr). Within array normalized data.	137
1.135MA plot of array 19 (slide35_062.gpr). Within array normalized data.	138
1.136Histogram of the array 19 (slide35_062.gpr). Within array normalized data.	139
1.137MA plot of array 20 (slide36_064.gpr). Within array normalized data.	140
1.138Histogram of the array 20 (slide36_064.gpr). Within array normalized data.	141
1.139MA plot of array 21 (slide37_063.gpr). Within array normalized data.	142
1.140Histogram of the array 21 (slide37_063.gpr). Within array normalized data.	143
1.141MA plot of array 22 (slide38_300.gpr). Within array normalized data.	144
1.142Histogram of the array 22 (slide38_300.gpr). Within array normalized data.	145
1.143MA plot of array 23 (slide39_072.gpr). Within array normalized data.	146
1.144Histogram of the array 23 (slide39_072.gpr). Within array normalized data.	147
1.145MA plot of array 24 (slide40_073.gpr). Within array normalized data.	148
1.146Histogram of the array 24 (slide40_073.gpr). Within array normalized data.	149
1.147Boxplots. Within array normalized data.	150

1.148Histogram of all arrays within this experiment before the between-array-normalization. The green lines corresponds to the green signal channels and the red line to the red channel respectively.	151
1.149MA plot of array 1 (slide01_056.gpr). Between array normalized data.	152
1.150Histogram of the array 1 (slide01_056.gpr). Between array normalized data.	153
1.151MA plot of array 2 (slide02_065.gpr). Between array normalized data.	154
1.152Histogram of the array 2 (slide02_065.gpr). Between array normalized data.	155
1.153MA plot of array 3 (slide03_058.gpr). Between array normalized data.	156
1.154Histogram of the array 3 (slide03_058.gpr). Between array normalized data.	157
1.155MA plot of array 4 (slide04_059.gpr). Between array normalized data.	158
1.156Histogram of the array 4 (slide04_059.gpr). Between array normalized data.	159
1.157MA plot of array 5 (slide05_060.gpr). Between array normalized data.	160
1.158Histogram of the array 5 (slide05_060.gpr). Between array normalized data.	161
1.159MA plot of array 6 (slide06_053.gpr). Between array normalized data.	162
1.160Histogram of the array 6 (slide06_053.gpr). Between array normalized data.	163
1.161MA plot of array 7 (slide07_054.gpr). Between array normalized data.	164
1.162Histogram of the array 7 (slide07_054.gpr). Between array normalized data.	165
1.163MA plot of array 8 (slide08_055.gpr). Between array normalized data.	166
1.164Histogram of the array 8 (slide08_055.gpr). Between array normalized data.	167
1.165MA plot of array 9 (slide17_095.gpr). Between array normalized data.	168
1.166Histogram of the array 9 (slide17_095.gpr). Between array normalized data.	169
1.167MA plot of array 10 (slide18_097.gpr). Between array normalized data.	170
1.168Histogram of the array 10 (slide18_097.gpr). Between array normalized data.	171
1.169MA plot of array 11 (slide19_098.gpr). Between array normalized data.	172
1.170Histogram of the array 11 (slide19_098.gpr). Between array normalized data.	173
1.171MA plot of array 12 (slide20_099.gpr). Between array normalized data.	174
1.172Histogram of the array 12 (slide20_099.gpr). Between array normalized data.	175
1.173MA plot of array 13 (slide21_045.gpr). Between array normalized data.	176
1.174Histogram of the array 13 (slide21_045.gpr). Between array normalized data.	177
1.175MA plot of array 14 (slide22_083.gpr). Between array normalized data.	178
1.176Histogram of the array 14 (slide22_083.gpr). Between array normalized data.	179
1.177MA plot of array 15 (slide23_298.gpr). Between array normalized data.	180
1.178Histogram of the array 15 (slide23_298.gpr). Between array normalized data.	181
1.179MA plot of array 16 (slide24_085.gpr). Between array normalized data.	182
1.180Histogram of the array 16 (slide24_085.gpr). Between array normalized data.	183

1.181MA plot of array 17 (slide33_100.gpr). Between array normalized data.	184
1.182Histogram of the array 17 (slide33_100.gpr). Between array normalized data.	185
1.183MA plot of array 18 (slide34_061.gpr). Between array normalized data.	186
1.184Histogram of the array 18 (slide34_061.gpr). Between array normalized data.	187
1.185MA plot of array 19 (slide35_062.gpr). Between array normalized data.	188
1.186Histogram of the array 19 (slide35_062.gpr). Between array normalized data.	189
1.187MA plot of array 20 (slide36_064.gpr). Between array normalized data.	190
1.188Histogram of the array 20 (slide36_064.gpr). Between array normalized data.	191
1.189MA plot of array 21 (slide37_063.gpr). Between array normalized data.	192
1.190Histogram of the array 21 (slide37_063.gpr). Between array normalized data.	193
1.191MA plot of array 22 (slide38_300.gpr). Between array normalized data.	194
1.192Histogram of the array 22 (slide38_300.gpr). Between array normalized data.	195
1.193MA plot of array 23 (slide39_072.gpr). Between array normalized data.	196
1.194Histogram of the array 23 (slide39_072.gpr). Between array normalized data.	197
1.195MA plot of array 24 (slide40_073.gpr). Between array normalized data.	198
1.196Histogram of the array 24 (slide40_073.gpr). Between array normalized data.	199
1.197Boxplots. Between array normalized data.	200
1.198Histogram of all arrays within this experiment after the between-array-normalization. The green lines corresponds to the green signal channels and the red line to the red channel respectively.	201
3.1 Mean vs standard deviation plot of the data.	206
3.2 Sorted variance.	207
3.3 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	209
3.4 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	210
3.5 MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.	211
3.6 Volcano plot scattering the average M values (x axis) against the raw p values (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	212

3.7 Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	213
4.1 Mean vs standard deviation plot of the data.	216
4.2 Sorted variance.	217
4.3 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	219
4.4 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	220
4.5 MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.	221
4.6 Volcano plot scattering the average M values (x axis) against the raw p values (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	222
4.7 Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	223
5.1 Mean vs standard deviation plot of the data.	226
5.2 Sorted variance.	227
5.3 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	229
5.4 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	230
5.5 MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.	231
5.6 Volcano plot scattering the average M values (x axis) against the raw p values (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	232

5.7 Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	233
6.1 Mean vs standard deviation plot of the data.	236
6.2 Sorted variance.	237
6.3 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	239
6.4 Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.	240
6.5 MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.	241
6.6 Volcano plot scattering the average M values (x axis) against the raw p values (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	242
6.7 Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.	243

List of Tables

Chapter 1

Preprocessing of two color microarrays

The data preprocessing of two color microarrays consists typically of the following steps:

- Background correction
- Normalizing within arrays
- Normalizing between arrays: adjust the expression / regulation values across all microarrays of the experiment.

For the analysis in this chapter the Bioconductor packages *limma* (normalization of the microarray data) and *maDB* (plotting functions) were used.

```
> library(limma)
> library(maDB)
> library(RColorBrewer)
> source("utils.R")
```

1.1 Reading the raw data

In this section the raw data files from the microarrays are read.

The experiment consists of the following microarrays:

- slide01_056.gpr
- slide02_065.gpr
- slide03_058.gpr

- slide04_059.gpr
- slide05_060.gpr
- slide06_053.gpr
- slide07_054.gpr
- slide08_055.gpr
- slide17_095.gpr
- slide18_097.gpr
- slide19_098.gpr
- slide20_099.gpr
- slide21_045.gpr
- slide22_083.gpr
- slide23_298.gpr
- slide24_085.gpr
- slide33_100.gpr
- slide34_061.gpr
- slide35_062.gpr
- slide36_064.gpr
- slide37_063.gpr
- slide38_300.gpr
- slide39_072.gpr
- slide40_073.gpr

```
> Slides.raw <- read.maimages(files = c("slide01_056.gpr", "slide02_065.gpr",
+   "slide03_058.gpr", "slide04_059.gpr", "slide05_060.gpr",
+   "slide06_053.gpr", "slide07_054.gpr", "slide08_055.gpr",
+   "slide17_095.gpr", "slide18_097.gpr", "slide19_098.gpr",
+   "slide20_099.gpr", "slide21_045.gpr", "slide22_083.gpr",
+   "slide23_298.gpr", "slide24_085.gpr", "slide33_100.gpr",
+   "slide34_061.gpr", "slide35_062.gpr", "slide36_064.gpr",
```

```

+   "slide37_063.gpr", "slide38_300.gpr", "slide39_072.gpr",
+   "slide40_073.gpr"), names = c("slide01_056.gpr", "slide02_065.gpr",
+   "slide03_058.gpr", "slide04_059.gpr", "slide05_060.gpr",
+   "slide06_053.gpr", "slide07_054.gpr", "slide08_055.gpr",
+   "slide17_095.gpr", "slide18_097.gpr", "slide19_098.gpr",
+   "slide20_099.gpr", "slide21_045.gpr", "slide22_083.gpr",
+   "slide23_298.gpr", "slide24_085.gpr", "slide33_100.gpr",
+   "slide34_061.gpr", "slide35_062.gpr", "slide36_064.gpr",
+   "slide37_063.gpr", "slide38_300.gpr", "slide39_072.gpr",
+   "slide40_073.gpr"), source = "genepix", sep = "\t", wt.fun = wtflags(0),
+   columns = list(Rf = "F635 Median", Gf = "F532 Median", Rb = "B635 Median",
+   Gb = "B532 Median"))

```

Read slide01_056.gpr
 Read slide02_065.gpr
 Read slide03_058.gpr
 Read slide04_059.gpr
 Read slide05_060.gpr
 Read slide06_053.gpr
 Read slide07_054.gpr
 Read slide08_055.gpr
 Read slide17_095.gpr
 Read slide18_097.gpr
 Read slide19_098.gpr
 Read slide20_099.gpr
 Read slide21_045.gpr
 Read slide22_083.gpr
 Read slide23_298.gpr
 Read slide24_085.gpr
 Read slide33_100.gpr
 Read slide34_061.gpr
 Read slide35_062.gpr
 Read slide36_064.gpr
 Read slide37_063.gpr
 Read slide38_300.gpr
 Read slide39_072.gpr
 Read slide40_073.gpr

1.2 Background correction

Next diagnostic plots of the raw data will be drawn. In an MA plot the regulation values (M values, differential expression) of all gene are plotted against their average expression values (A). Genes flagged by the scanning software are usually not drawn (provided that the exclusion of flagged genes was selected). The red and green lines in the plot represent the mean and median M respectiveley A value. The turquoise line in the plot represents the lowess fit line.

```
> plotHistogramFromSlide(Slides.raw, slide = 1, lwd = 2, log.transform = TRUE)
```

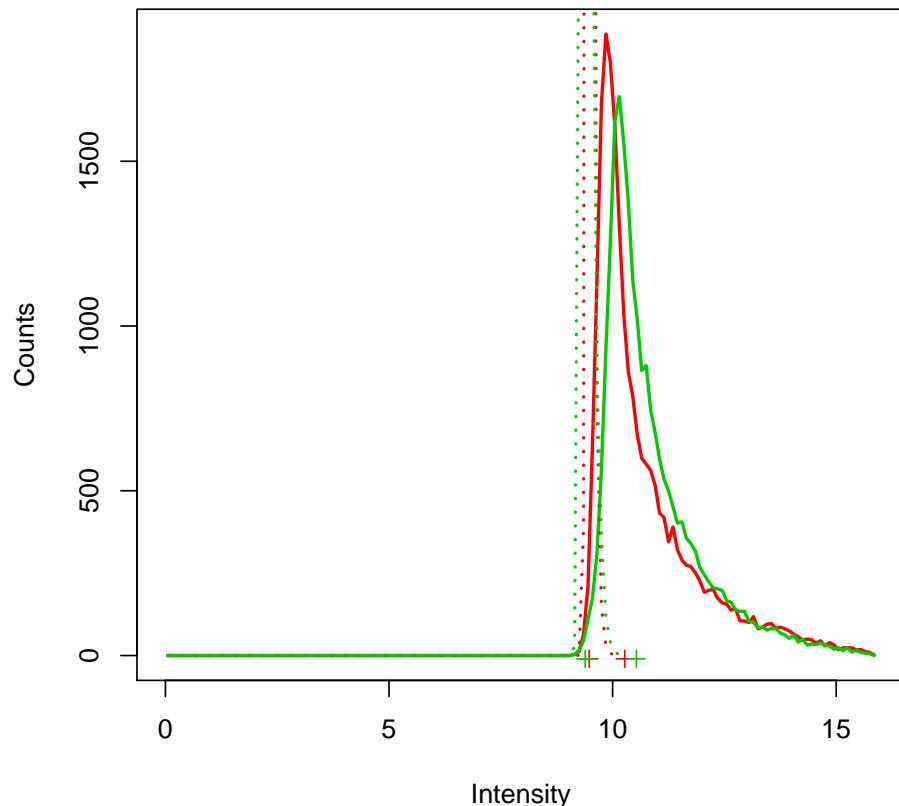
slide01_056.gpr

Figure 1.1: Histogram of the array 1 (slide01_056.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 2, lwd = 2, log.transform = TRUE)
```

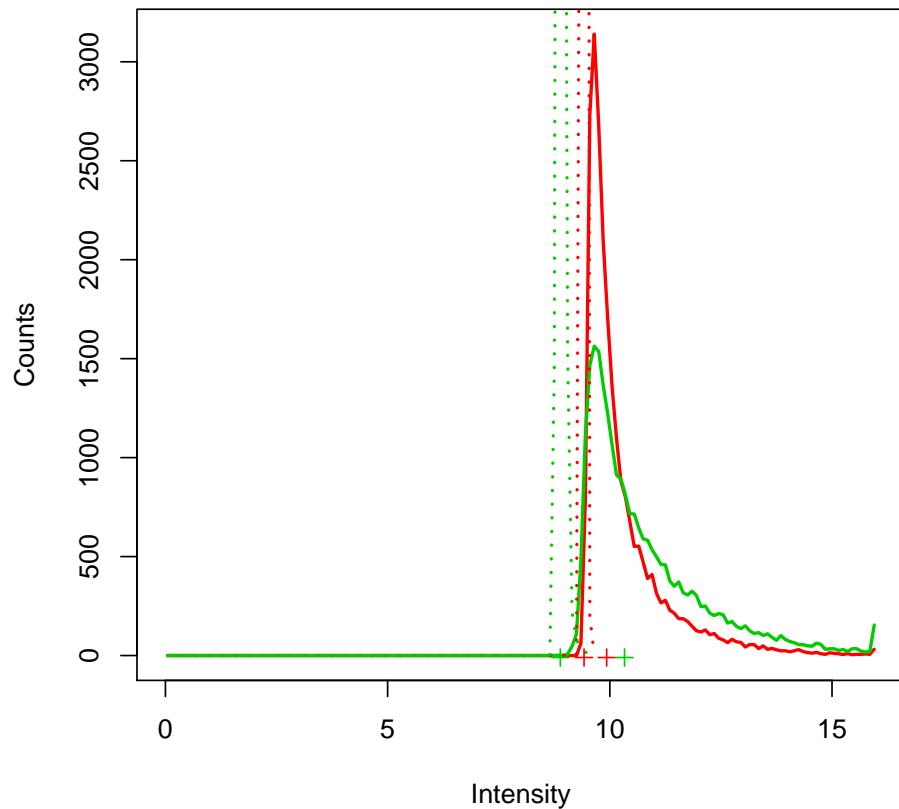
slide02_065.gpr

Figure 1.2: Histogram of the array 2 (slide02_065.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 3, lwd = 2, log.transform = TRUE)
```

slide03_058.gpr

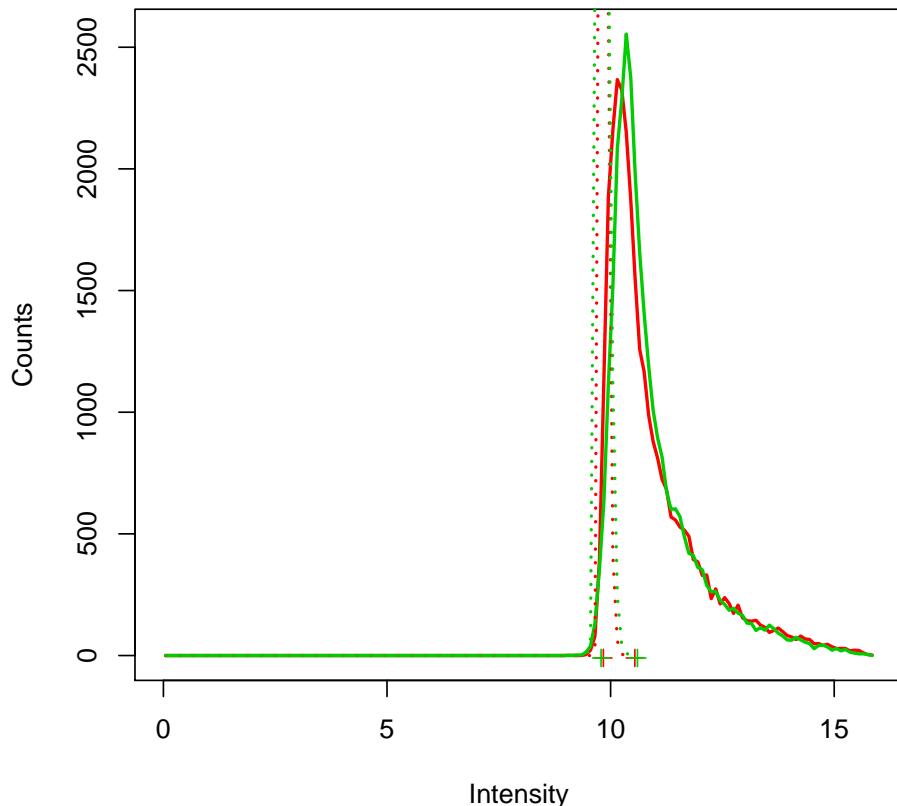


Figure 1.3: Histogram of the array 3 (slide03_058.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 4, lwd = 2, log.transform = TRUE)
```

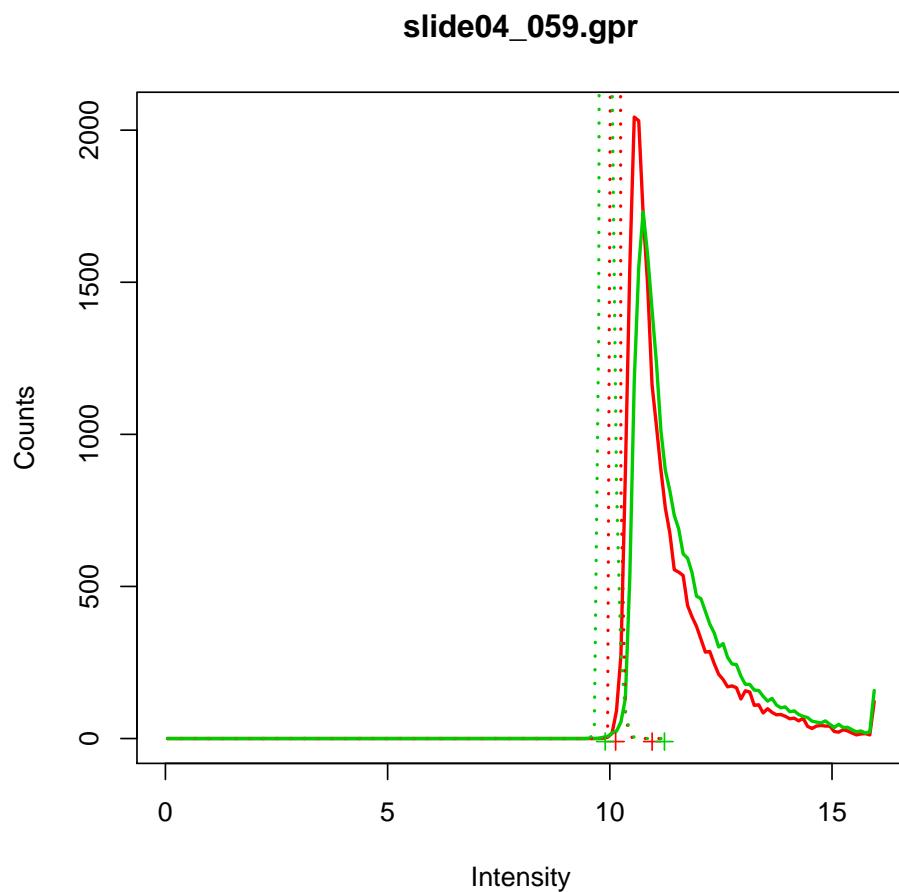


Figure 1.4: Histogram of the array 4 (slide04_059.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 5, lwd = 2, log.transform = TRUE)
```

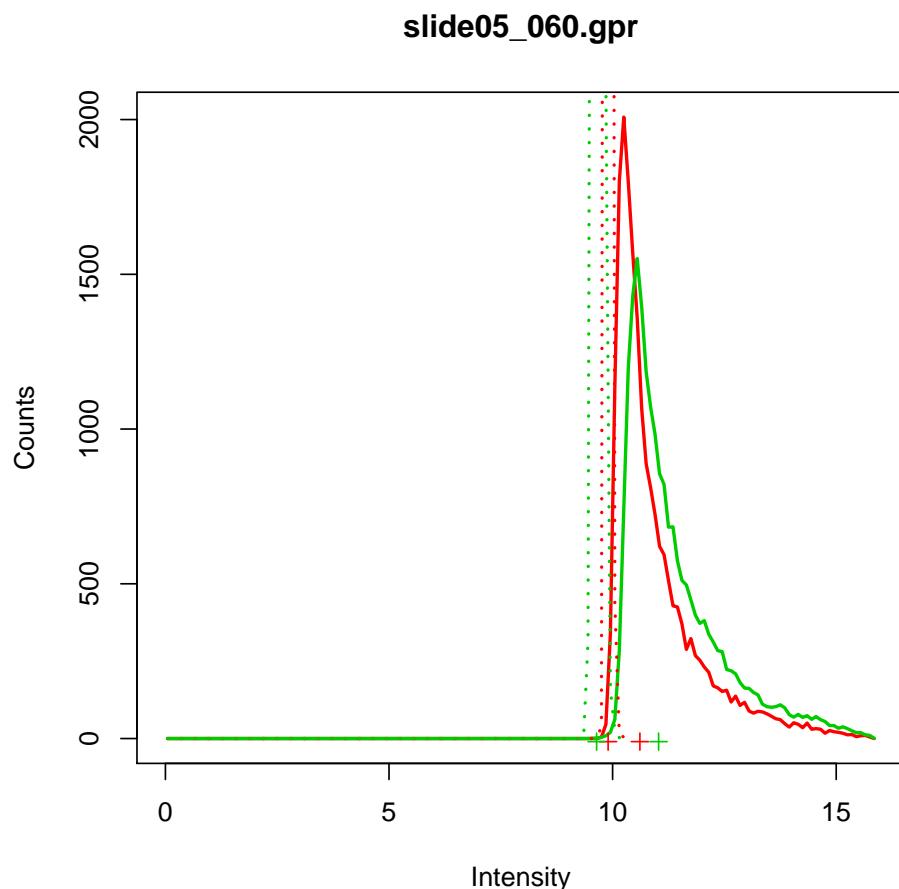


Figure 1.5: Histogram of the array 5 (slide05_060.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 6, lwd = 2, log.transform = TRUE)
```

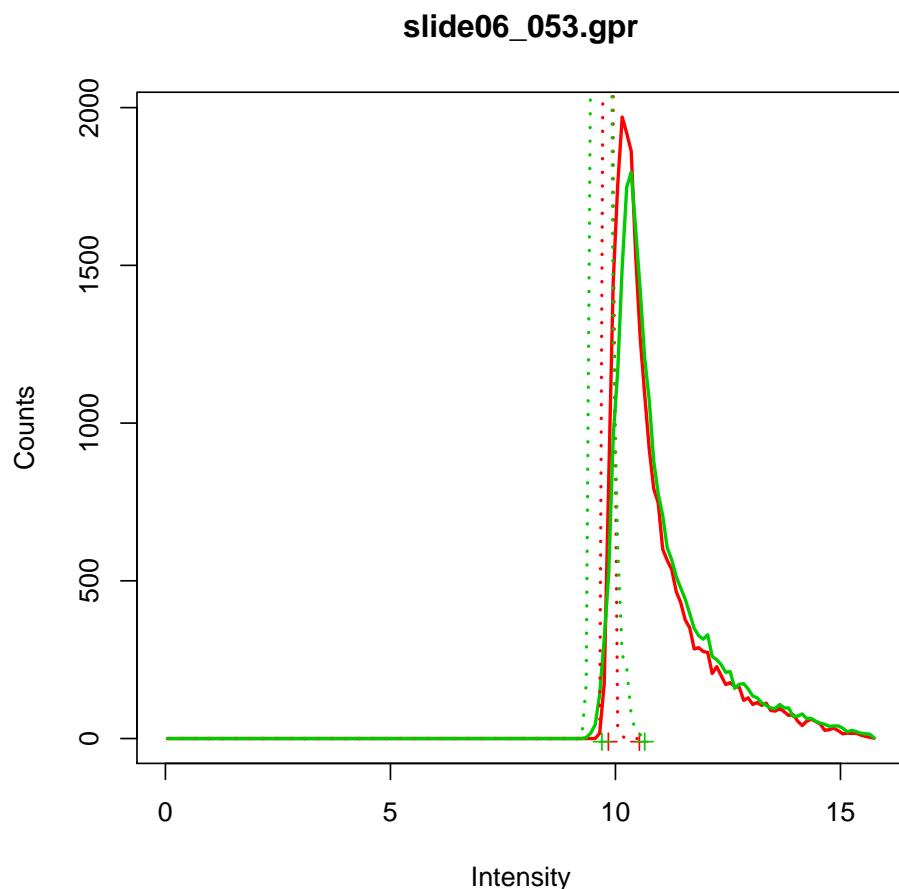


Figure 1.6: Histogram of the array 6 (slide06_053.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 7, lwd = 2, log.transform = TRUE)
```

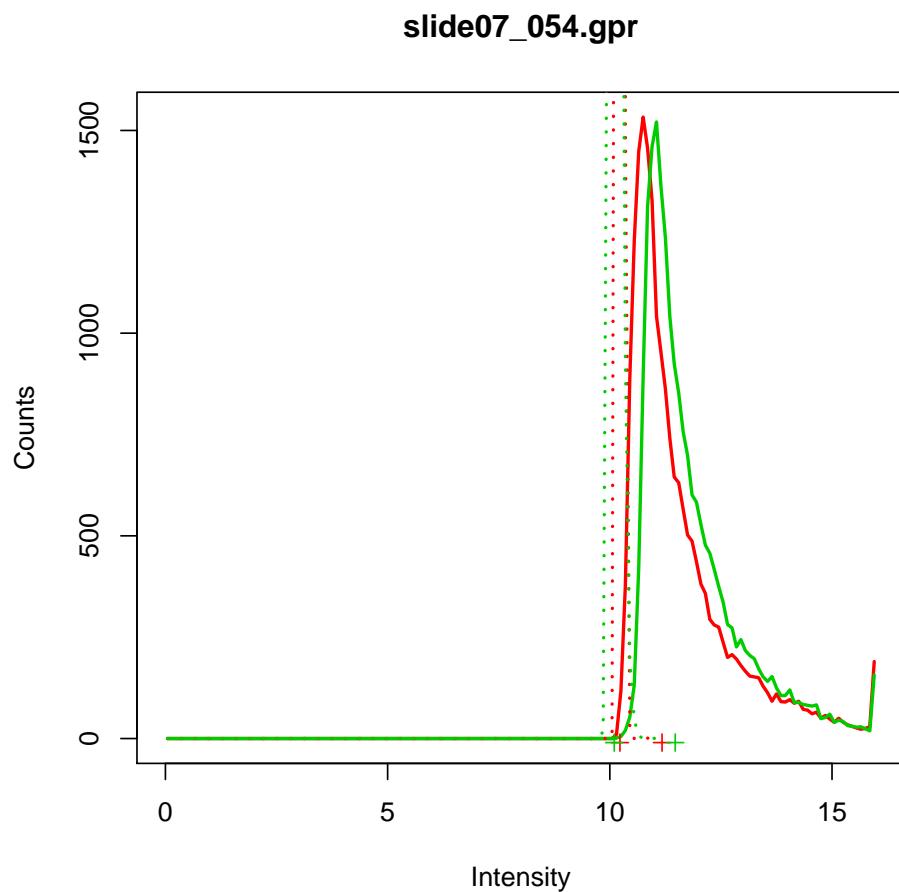


Figure 1.7: Histogram of the array 7 (slide07_054.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 8, lwd = 2, log.transform = TRUE)
```

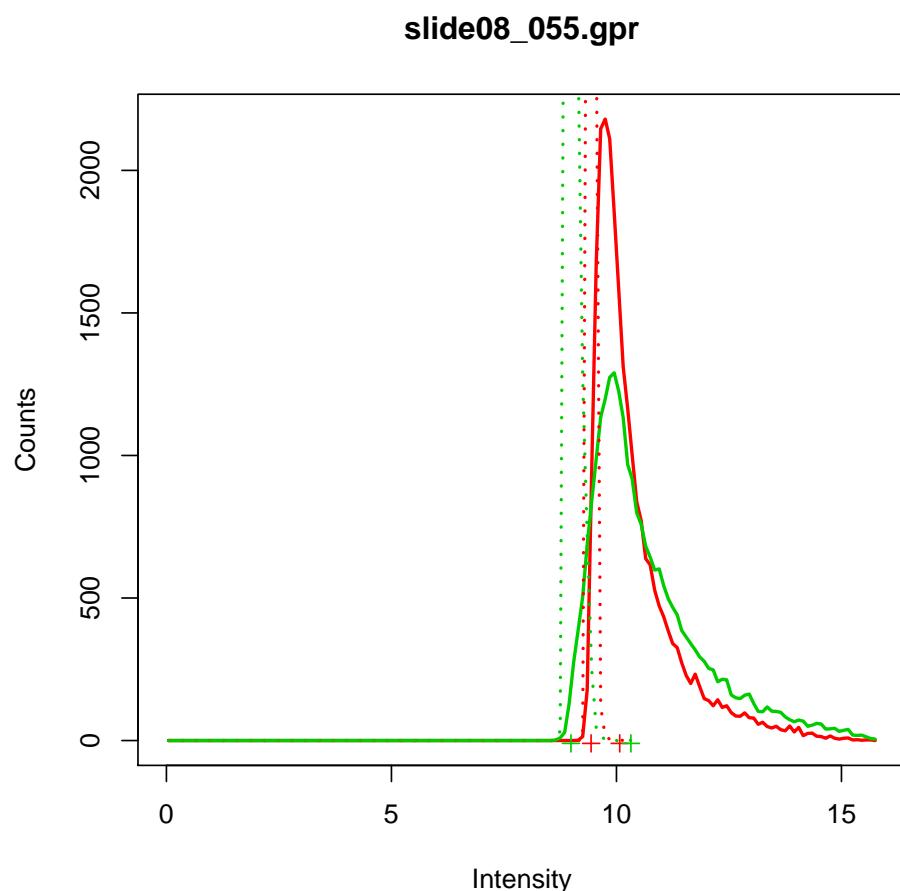


Figure 1.8: Histogram of the array 8 (slide08_055.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 9, lwd = 2, log.transform = TRUE)
```

slide17_095.gpr

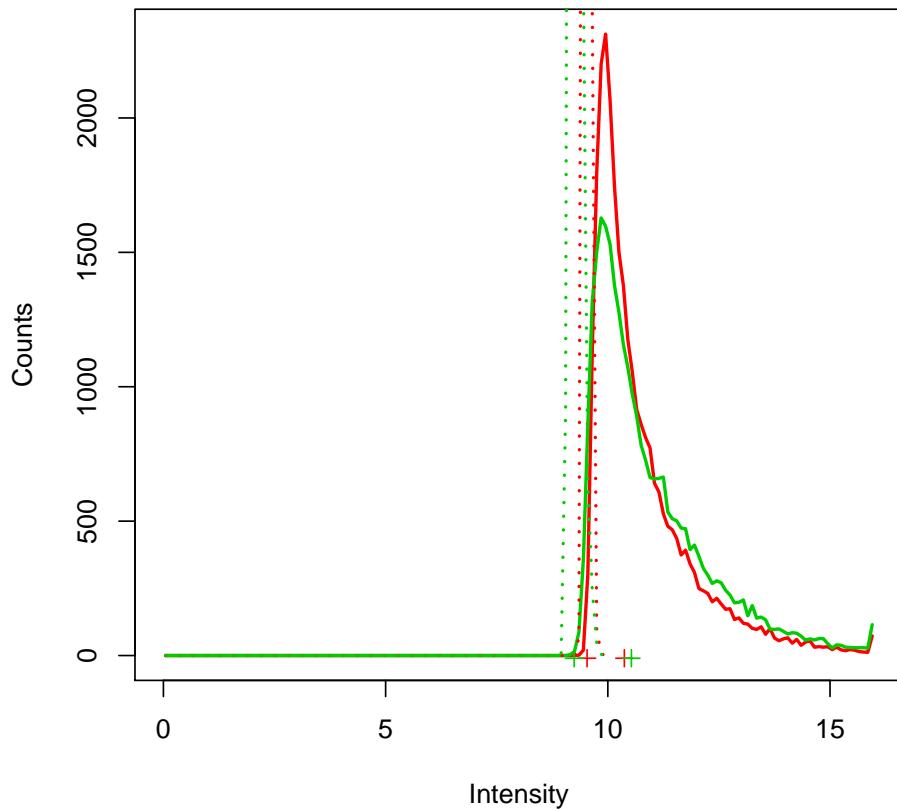


Figure 1.9: Histogram of the array 9 (slide17_095.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 10, lwd = 2, log.transform = TRUE)
```

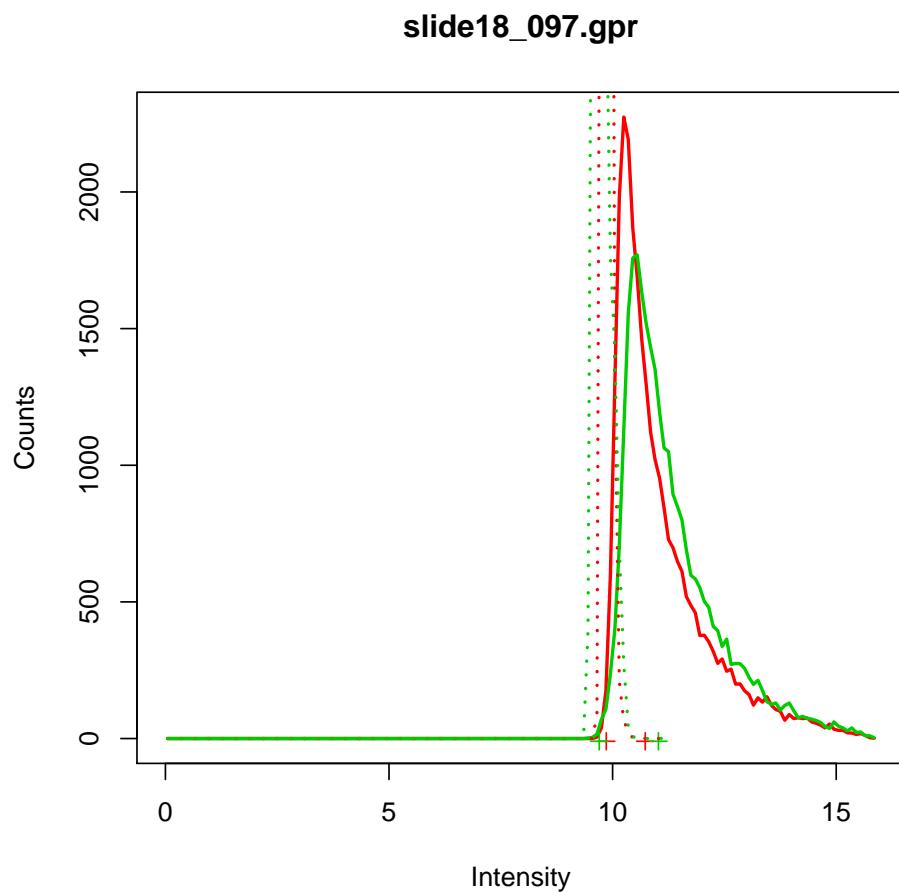


Figure 1.10: Histogram of the array 10 (slide18_097.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 11, lwd = 2, log.transform = TRUE)
```

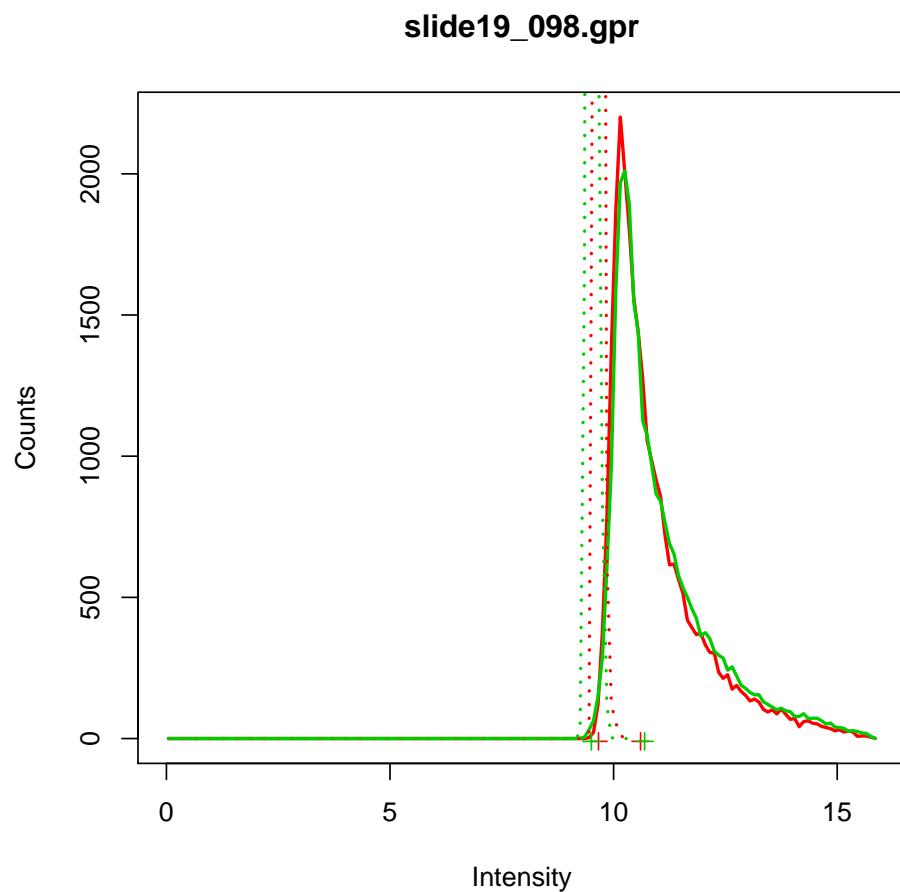


Figure 1.11: Histogram of the array 11 (slide19_098.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 12, lwd = 2, log.transform = TRUE)
```

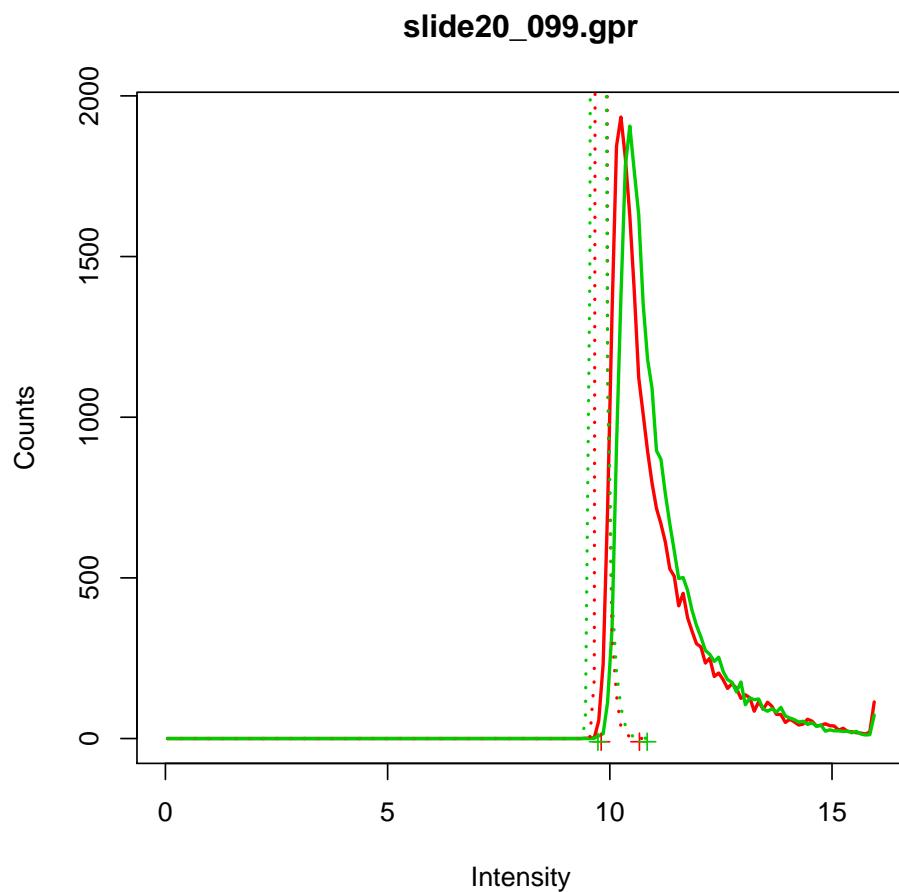


Figure 1.12: Histogram of the array 12 (slide20_099.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 13, lwd = 2, log.transform = TRUE)
```

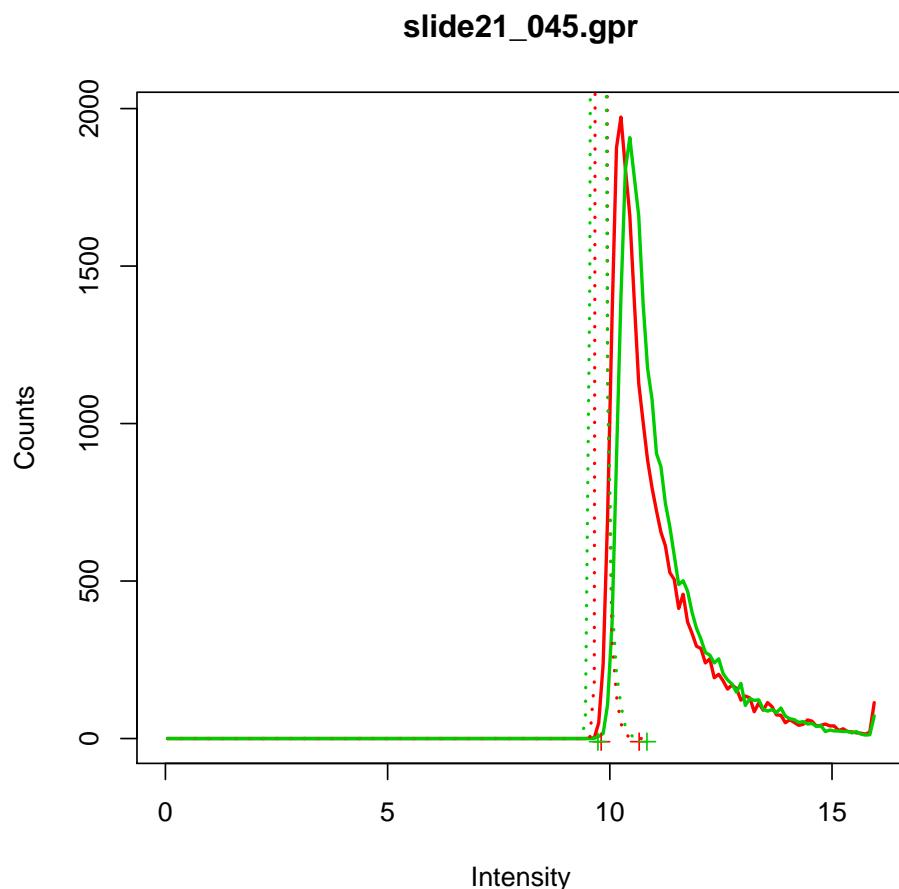


Figure 1.13: Histogram of the array 13 (slide21_045.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 14, lwd = 2, log.transform = TRUE)
```

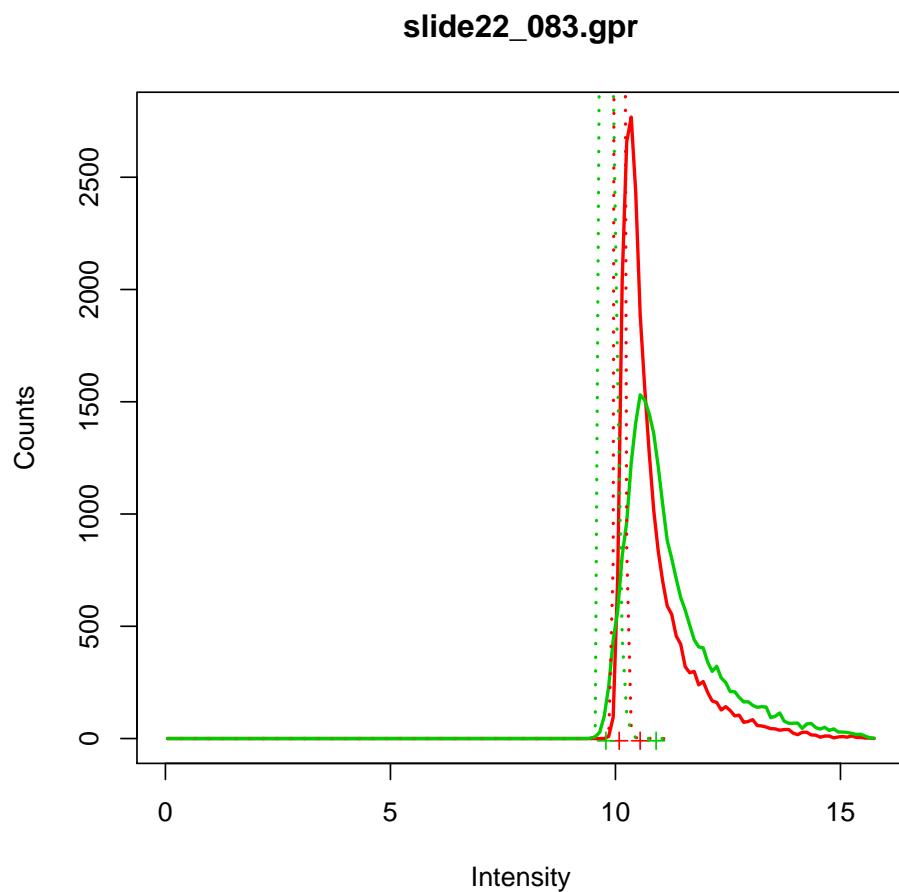


Figure 1.14: Histogram of the array 14 (slide22_083.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 15, lwd = 2, log.transform = TRUE)
```

slide23_298.gpr

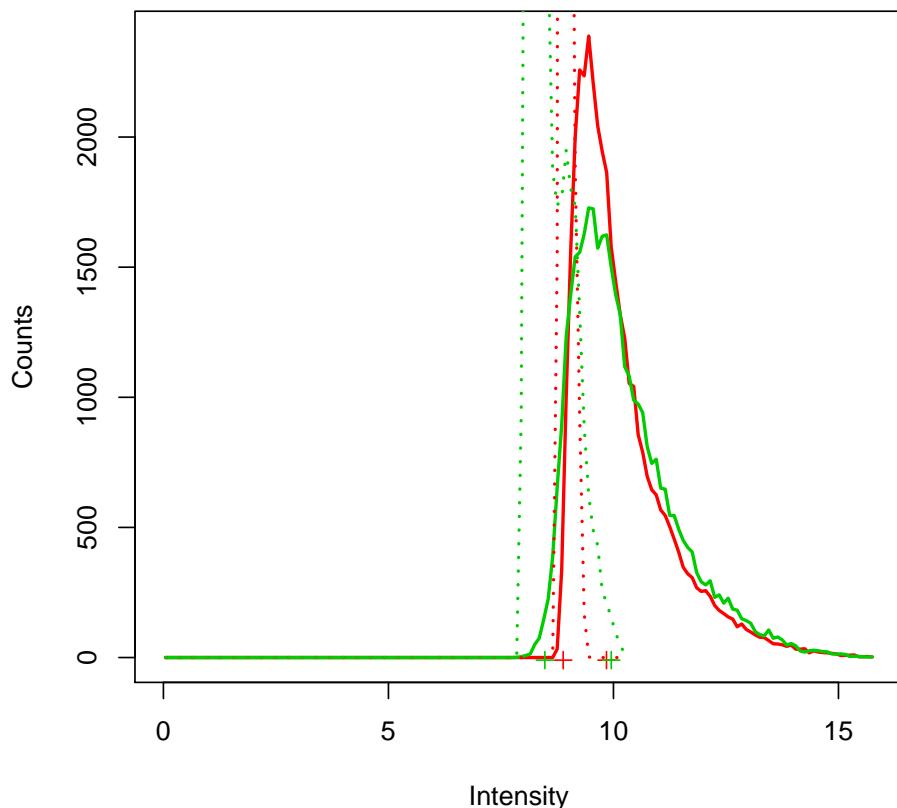


Figure 1.15: Histogram of the array 15 (slide23_298.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 16, lwd = 2, log.transform = TRUE)
```

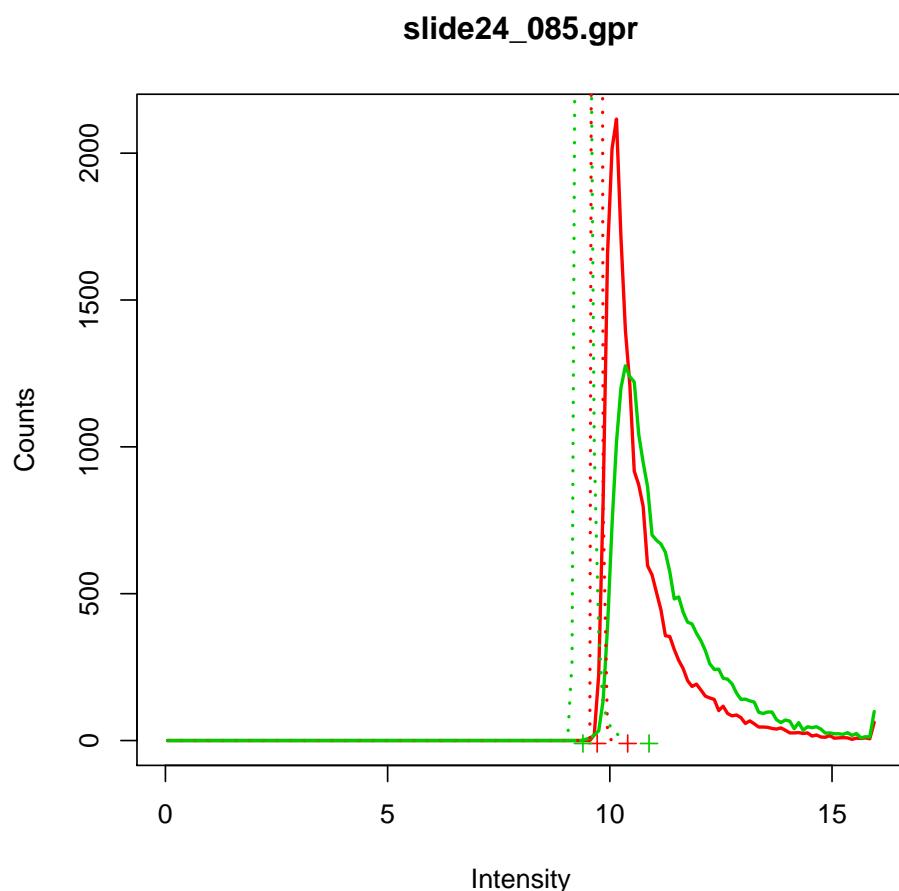


Figure 1.16: Histogram of the array 16 (slide24_085.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 17, lwd = 2, log.transform = TRUE)
```

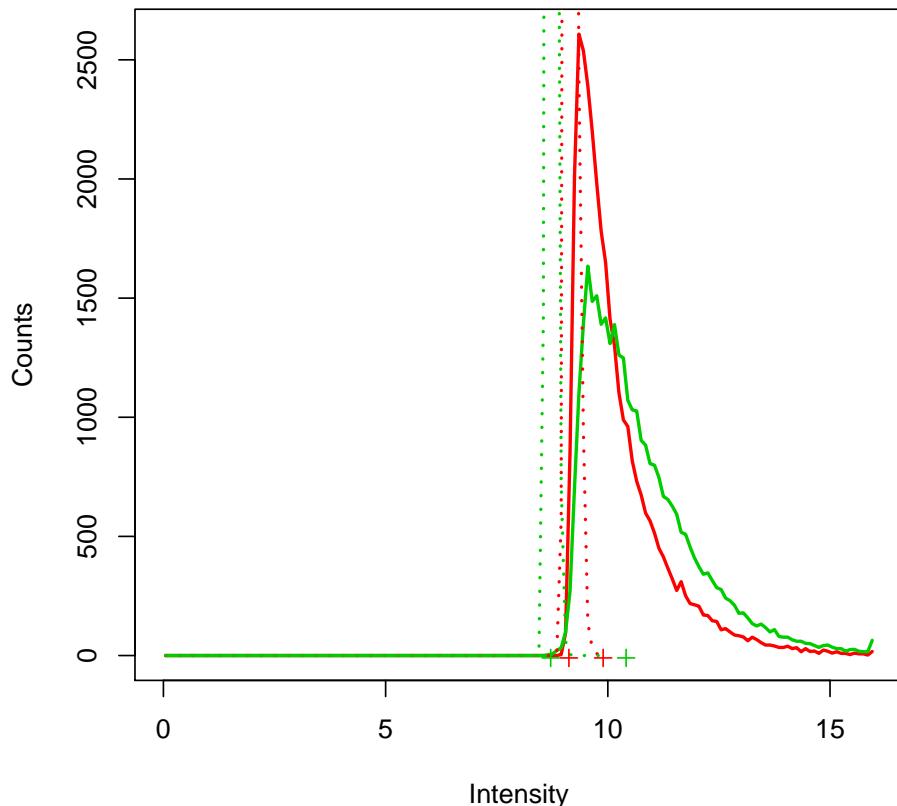
slide33_100.gpr

Figure 1.17: Histogram of the array 17 (slide33_100.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 18, lwd = 2, log.transform = TRUE)
```

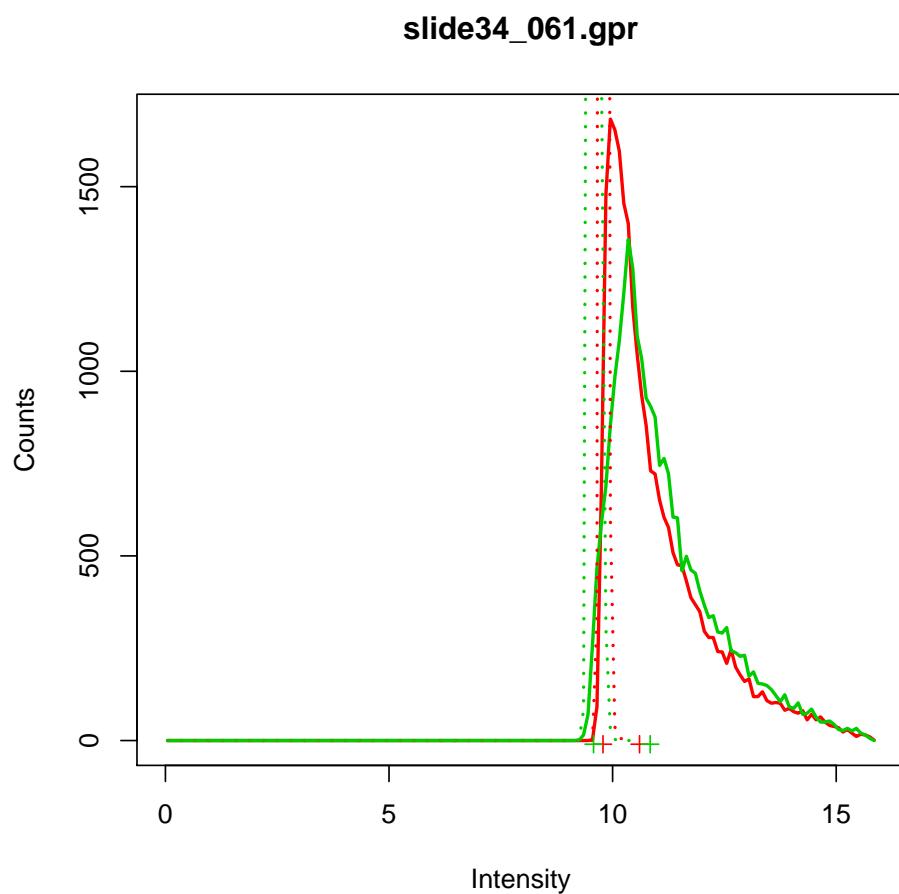


Figure 1.18: Histogram of the array 18 (slide34_061.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 19, lwd = 2, log.transform = TRUE)
```

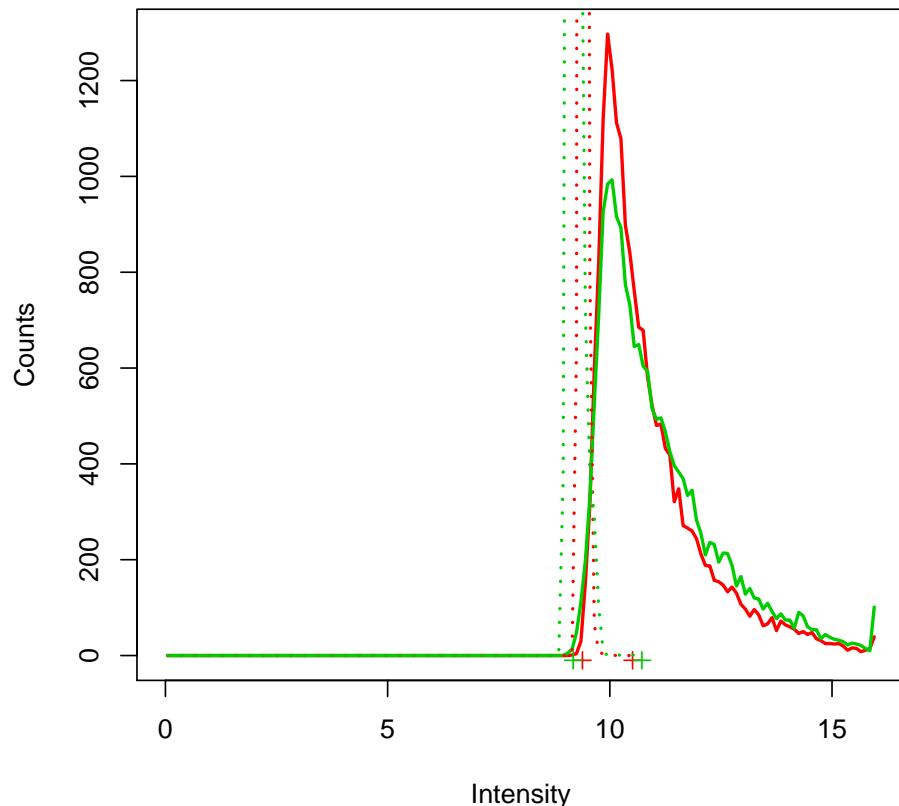
slide35_062.gpr

Figure 1.19: Histogram of the array 19 (slide35_062.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 20, lwd = 2, log.transform = TRUE)
```

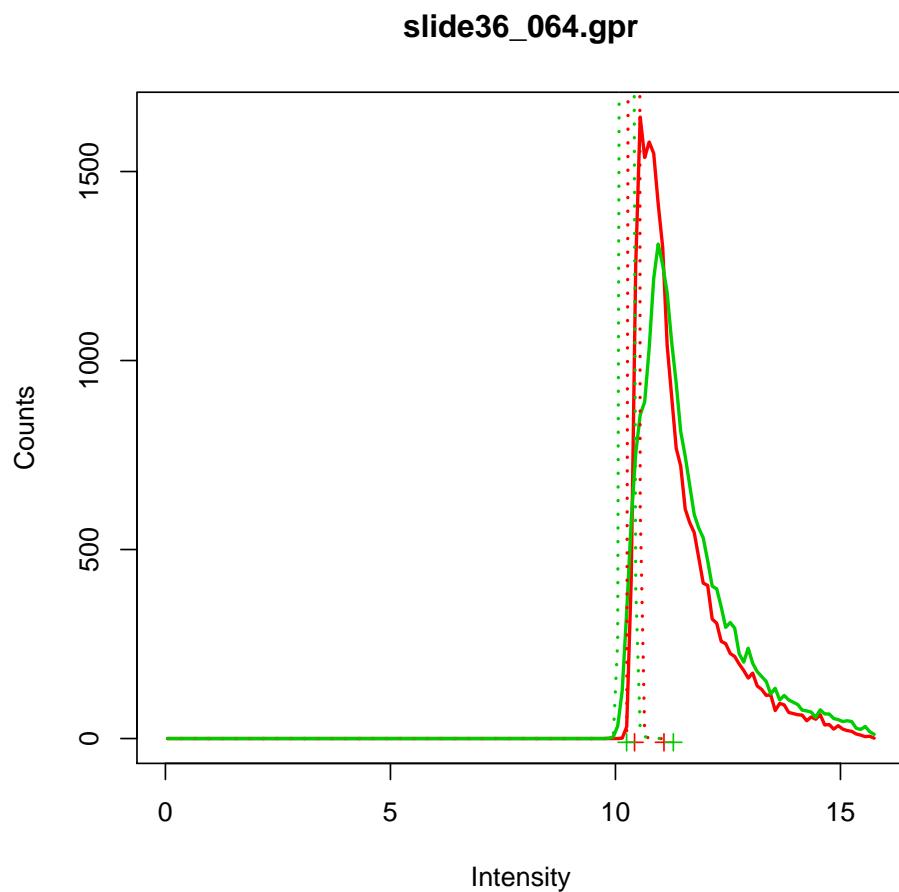


Figure 1.20: Histogram of the array 20 (slide36_064.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 21, lwd = 2, log.transform = TRUE)
```

slide37_063.gpr

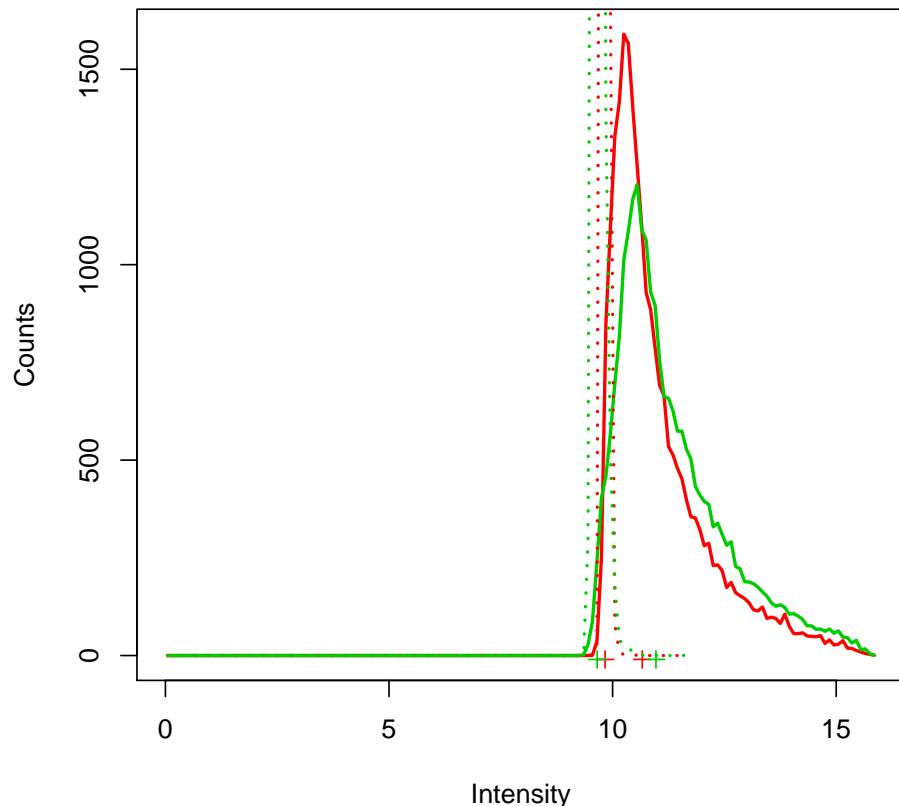


Figure 1.21: Histogram of the array 21 (slide37_063.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 22, lwd = 2, log.transform = TRUE)
```

slide38_300.gpr

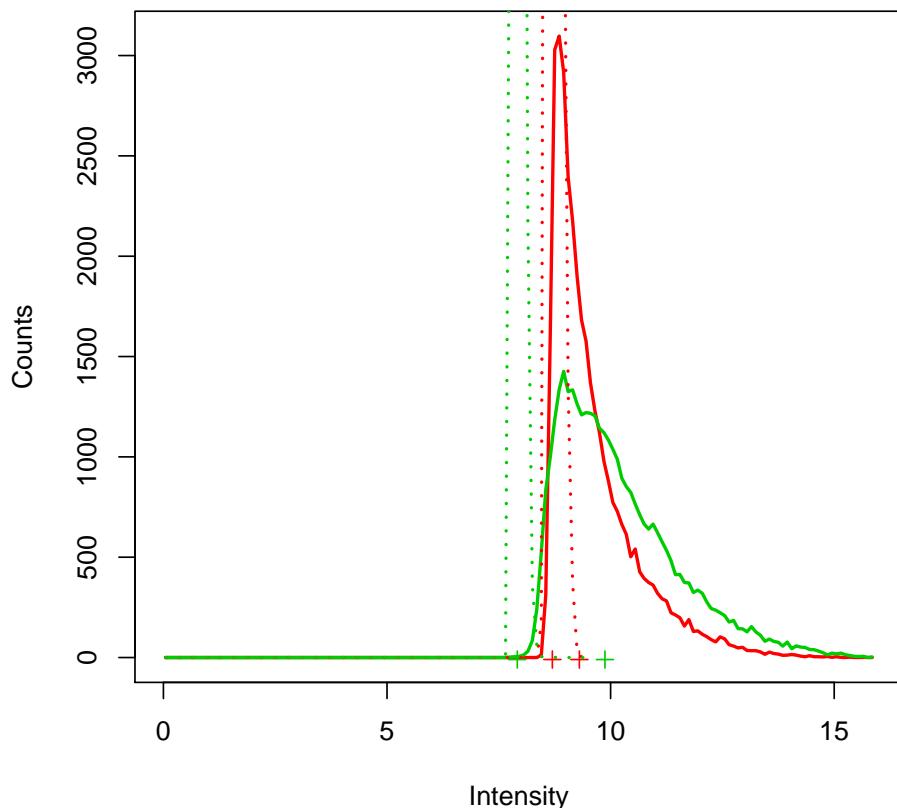


Figure 1.22: Histogram of the array 22 (slide38_300.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 23, lwd = 2, log.transform = TRUE)
```

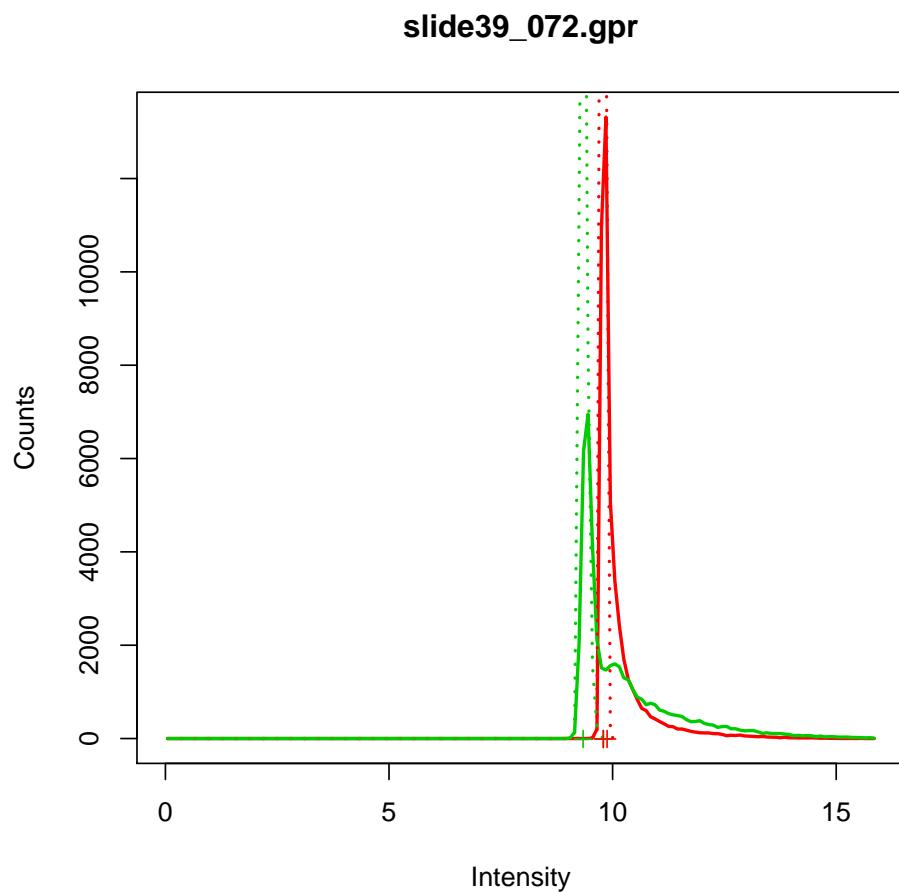


Figure 1.23: Histogram of the array 23 (slide39_072.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> plotHistogramFromSlide(Slides.raw, slide = 24, lwd = 2, log.transform = TRUE)
```

slide40_073.gpr

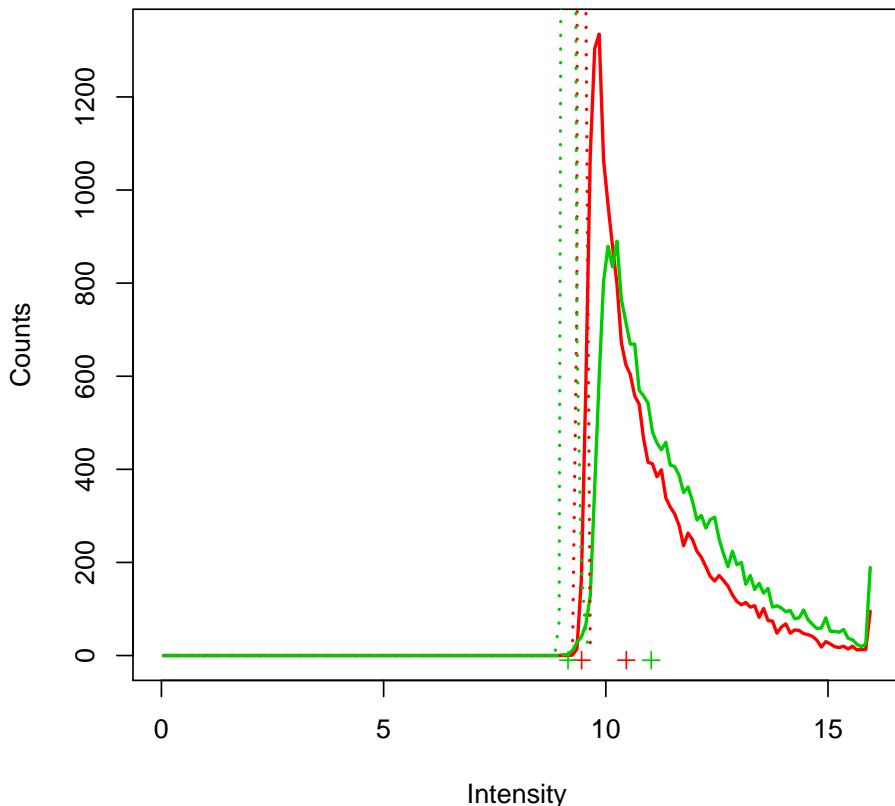


Figure 1.24: Histogram of the array 24 (slide40_073.gpr). Raw data before background correction. The green line corresponds to the green signal channel and the red line to the red channel. Dotted lines represent the background intensities.

```
> Dummy <- newMadbSet(Slides.raw)

Converting a limma RGLList into a MadbSet...
Setting the weights... a weights of 0 means the gene was flagged, a weights of one means the signal is ok!

Inserting available annotation into the slot @genes

Inserting available annotation into the slot @genes
```

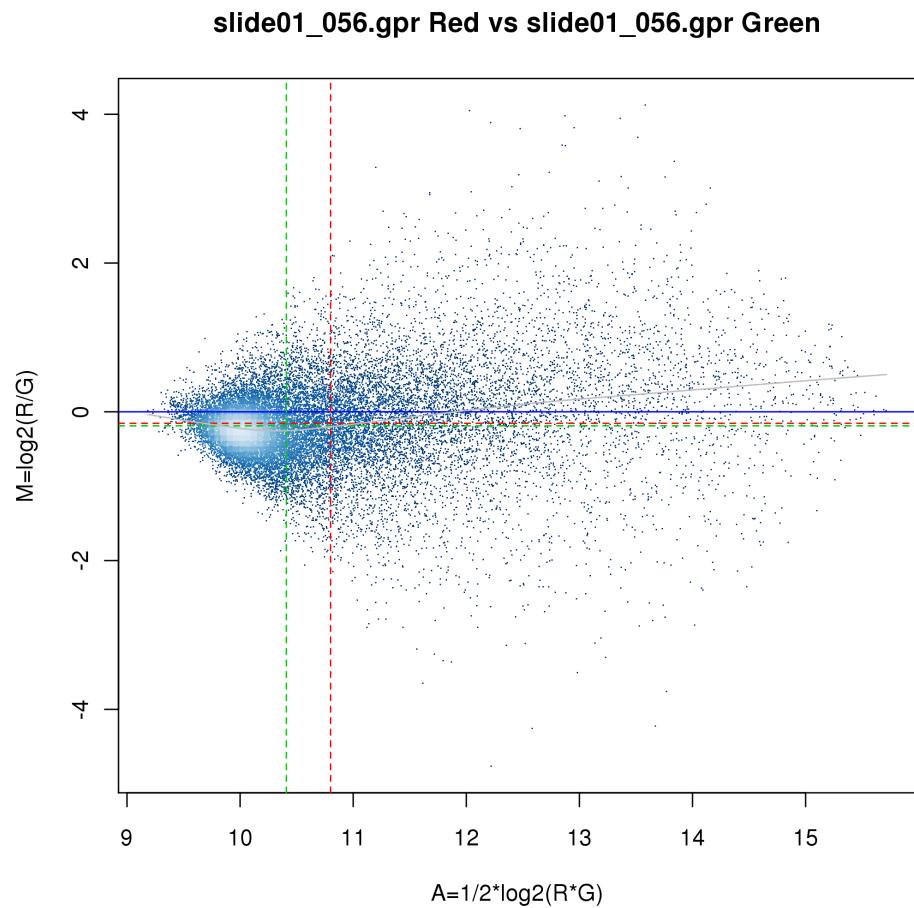


Figure 1.25: MA plot of array 1 (slide01_056.gpr). Raw data before background correction.

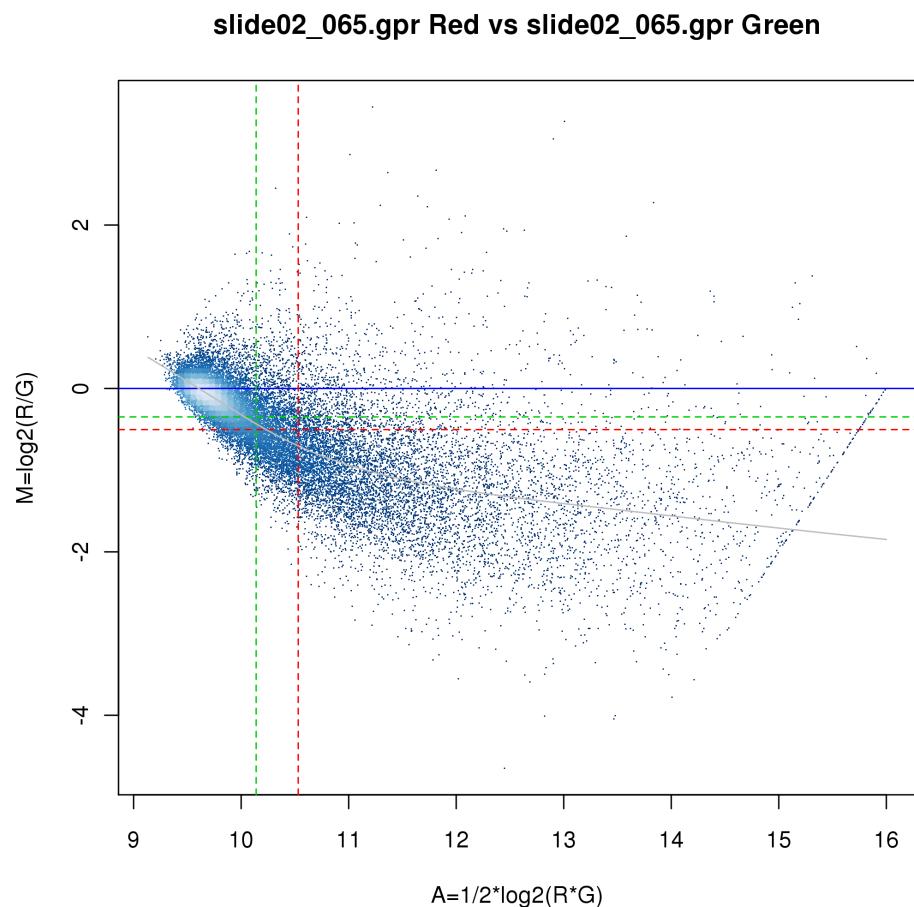


Figure 1.26: MA plot of array 2 (slide02_065.gpr). Raw data before background correction.

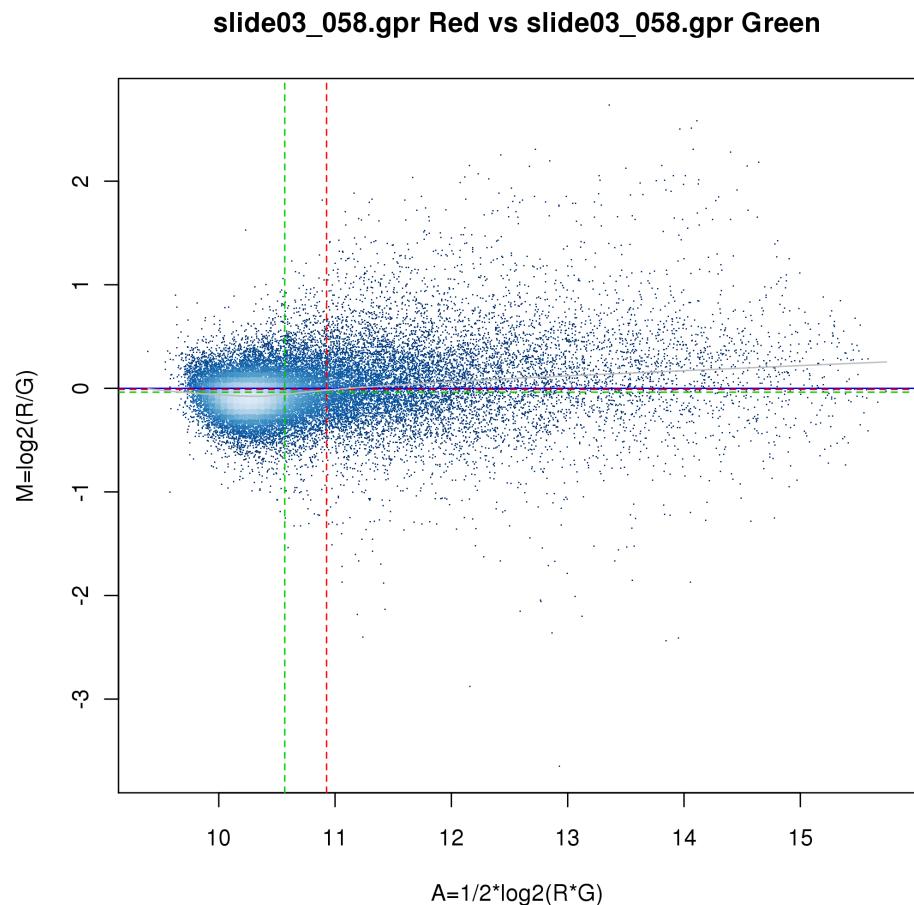


Figure 1.27: MA plot of array 3 (slide03_058.gpr). Raw data before background correction.

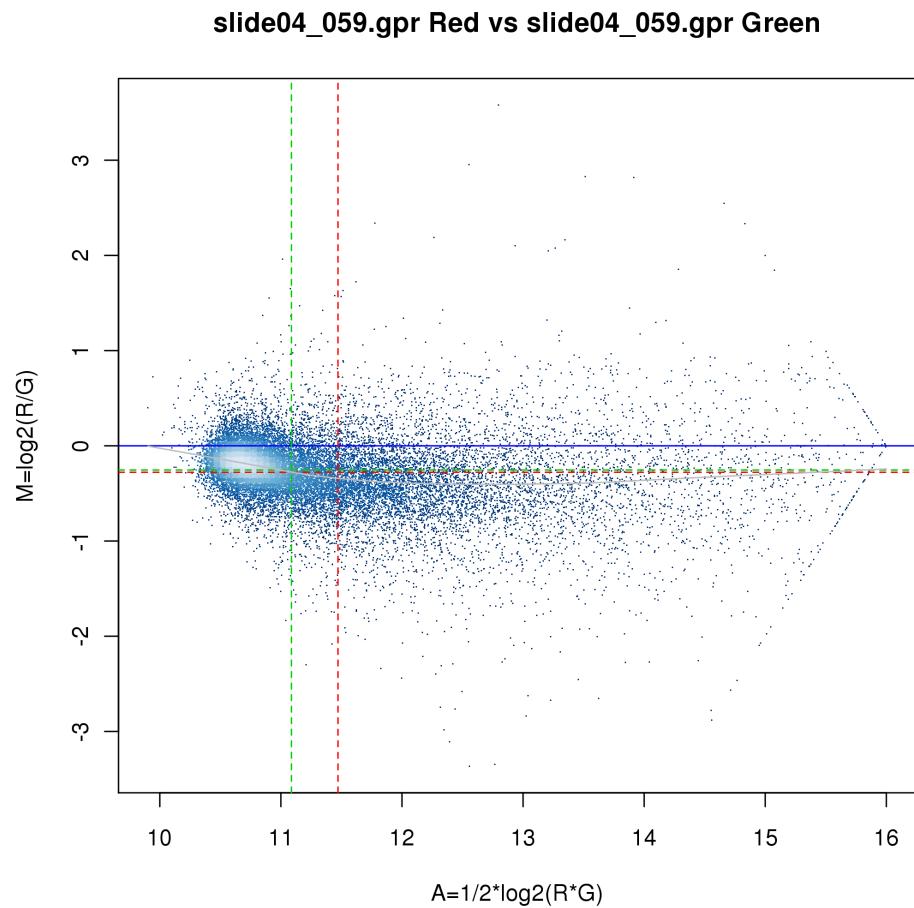


Figure 1.28: MA plot of array 4 (slide04_059.gpr). Raw data before background correction.

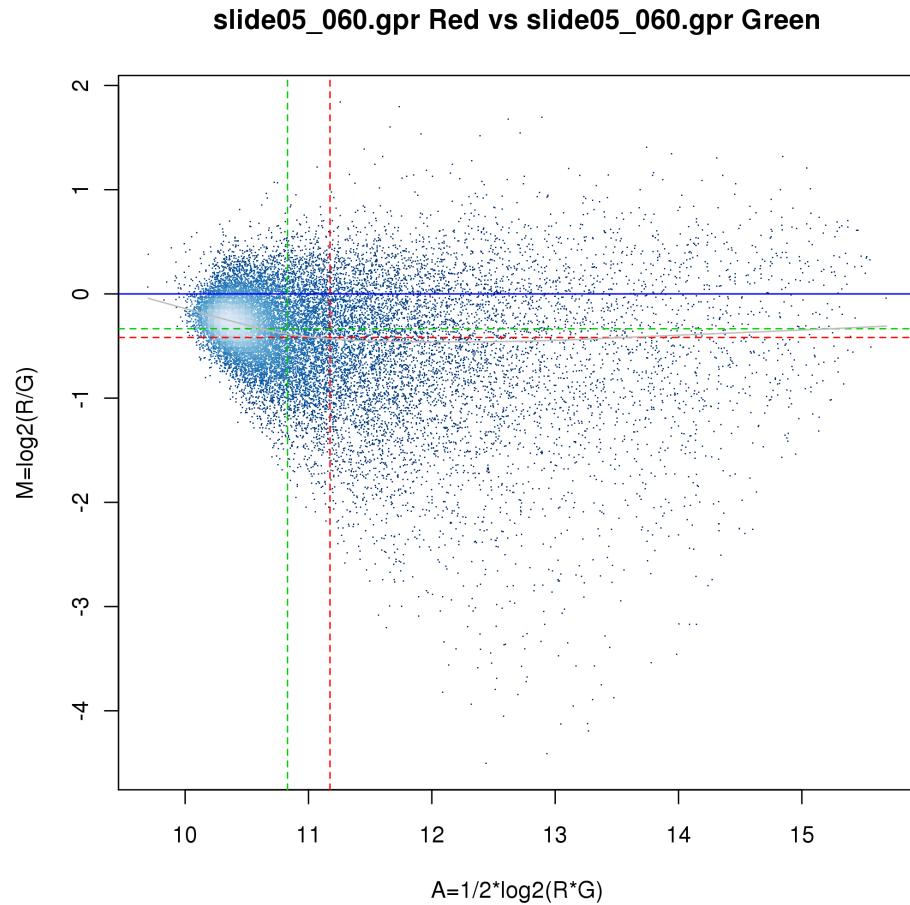


Figure 1.29: MA plot of array 5 (slide05_060.gpr). Raw data before background correction.

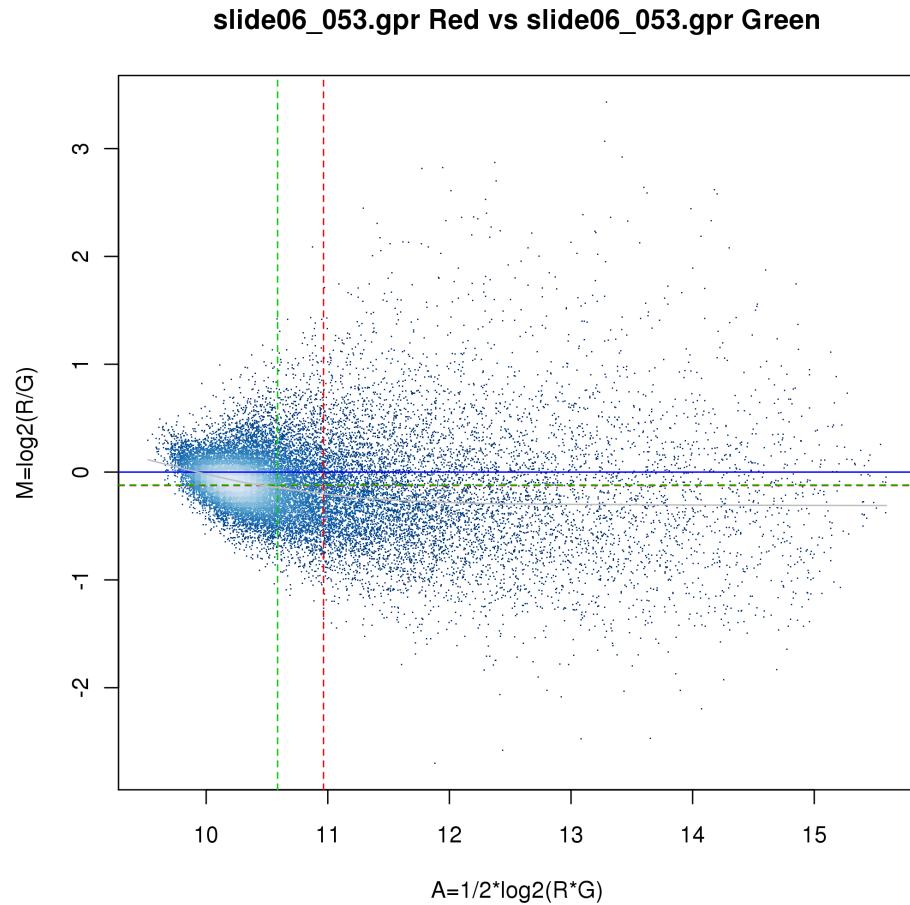


Figure 1.30: MA plot of array 6 (slide06_053.gpr). Raw data before background correction.

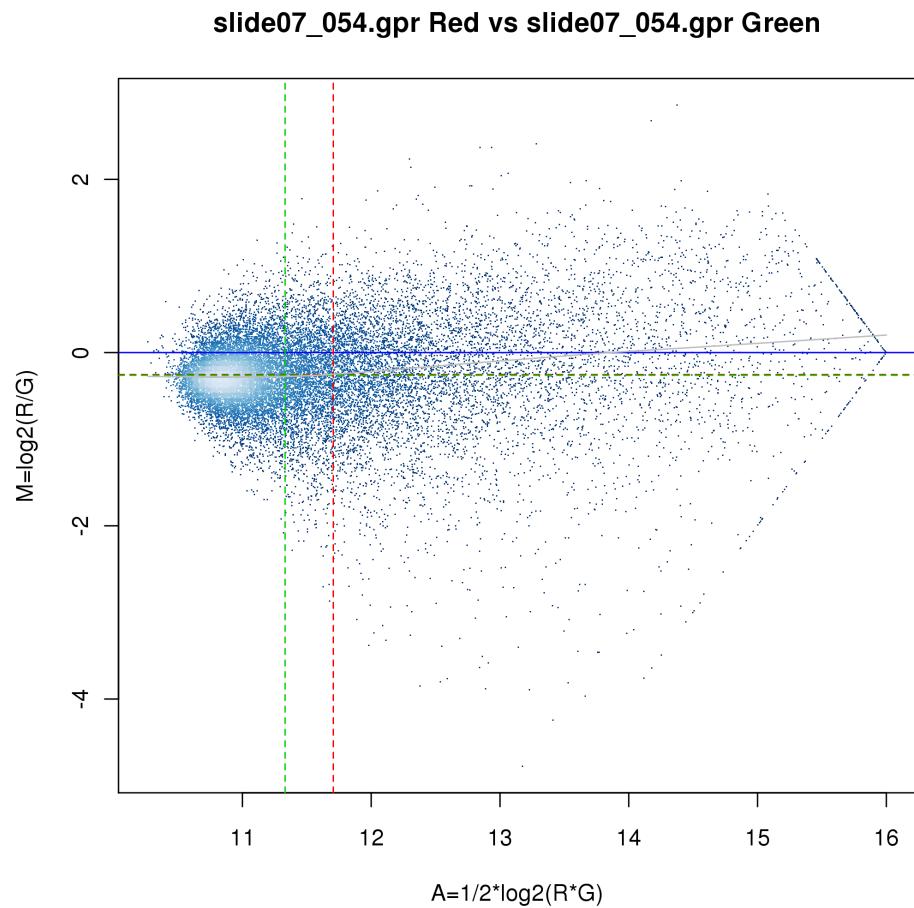


Figure 1.31: MA plot of array 7 (slide07_054.gpr). Raw data before background correction.

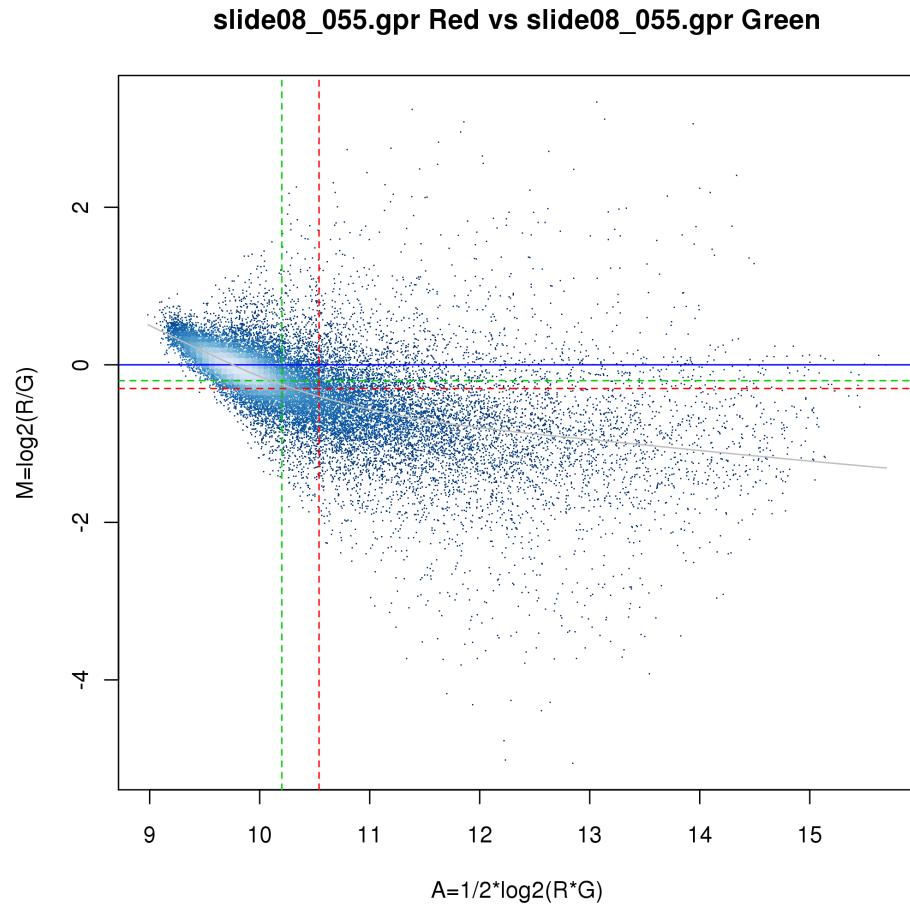


Figure 1.32: MA plot of array 8 (slide08_055.gpr). Raw data before background correction.

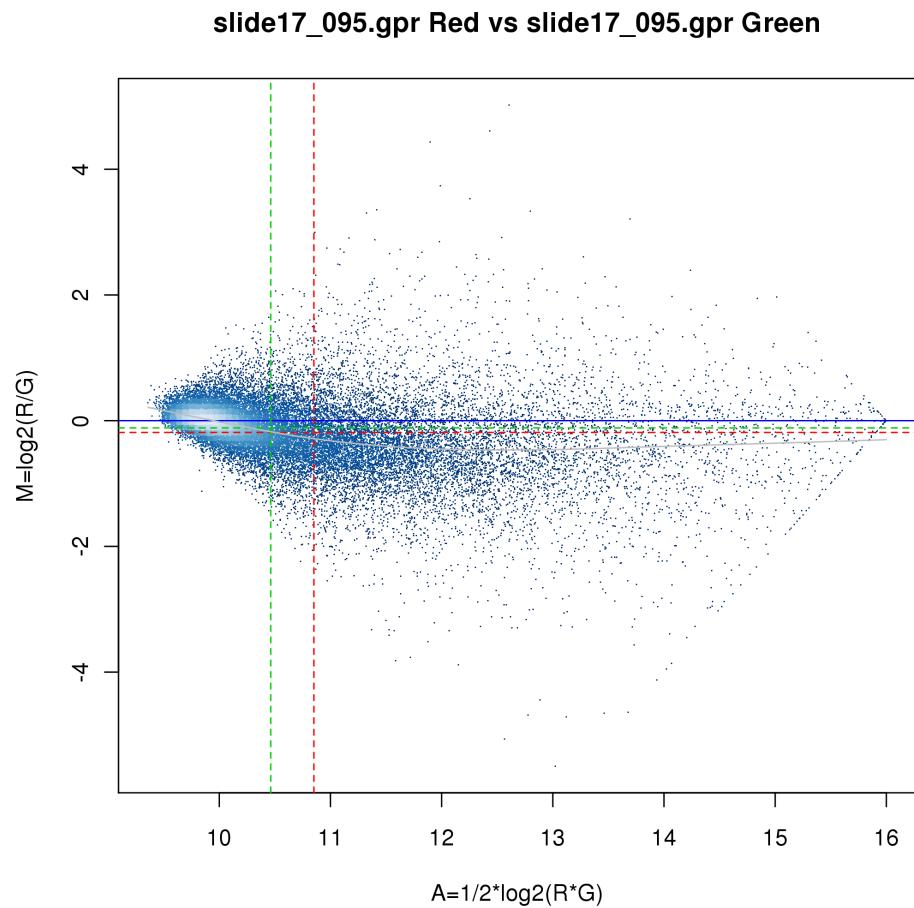


Figure 1.33: MA plot of array 9 (slide17_095.gpr). Raw data before background correction.

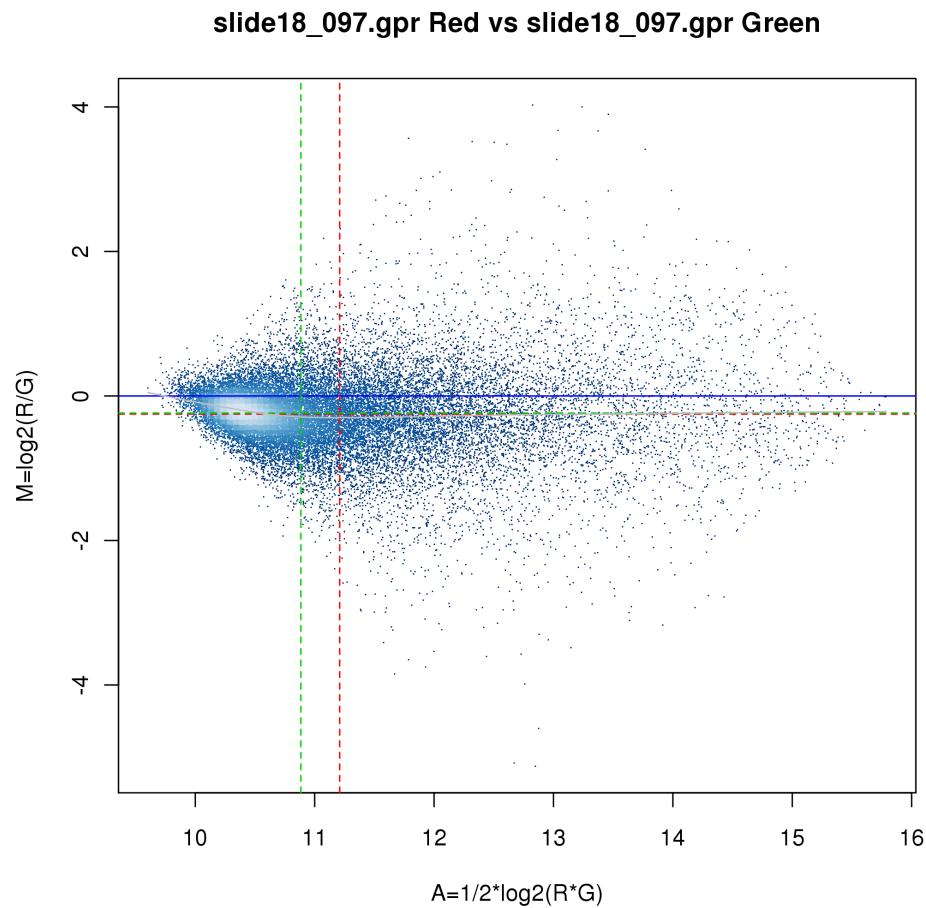


Figure 1.34: MA plot of array 10 (slide18_097.gpr). Raw data before background correction.

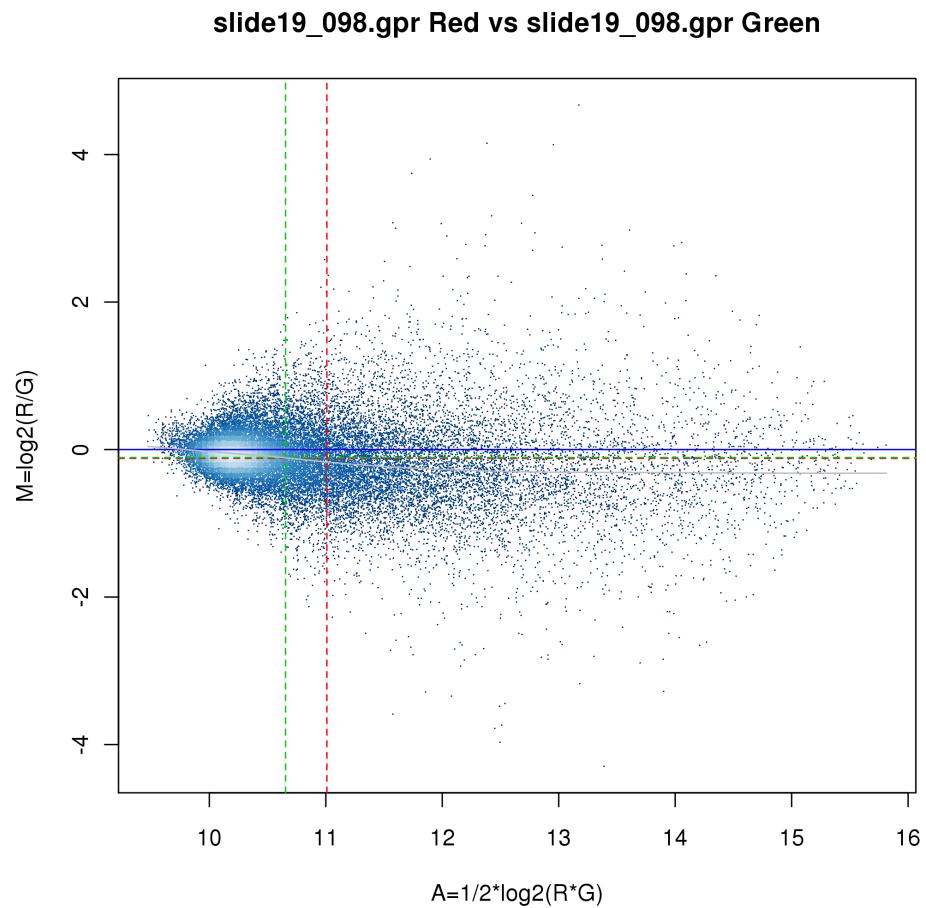


Figure 1.35: MA plot of array 11 (slide19_098.gpr). Raw data before background correction.

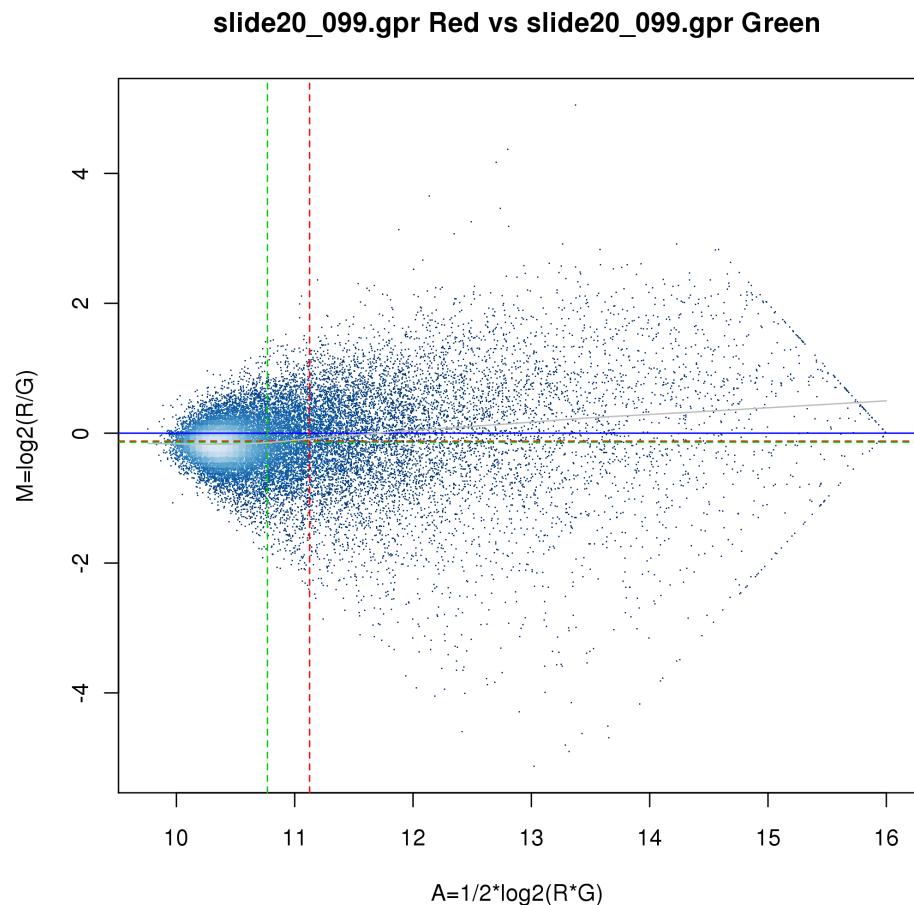


Figure 1.36: MA plot of array 12 (slide20_099.gpr). Raw data before background correction.

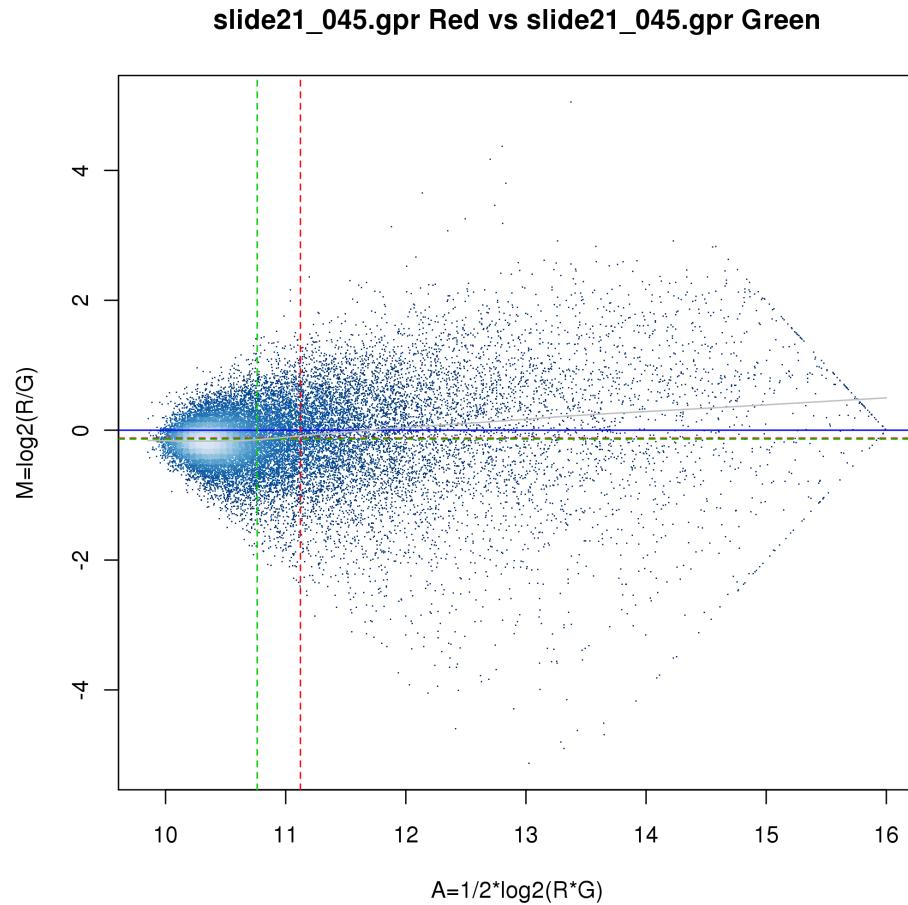


Figure 1.37: MA plot of array 13 (slide21_045.gpr). Raw data before background correction.

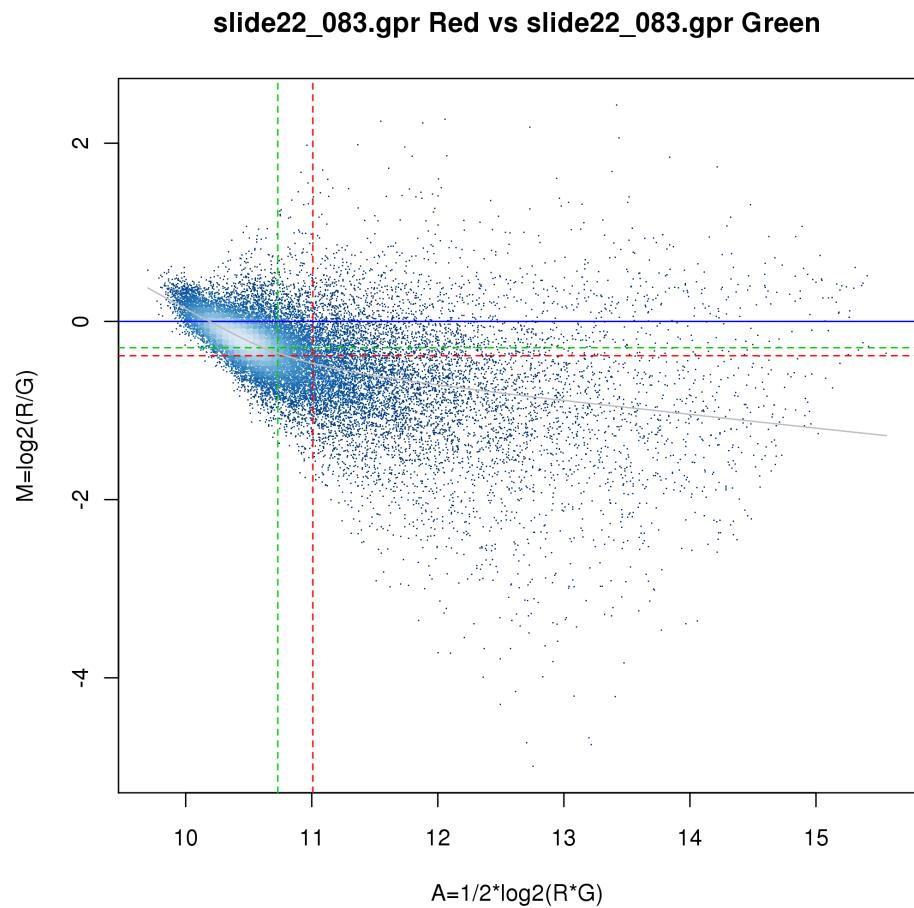


Figure 1.38: MA plot of array 14 (slide22_083.gpr). Raw data before background correction.

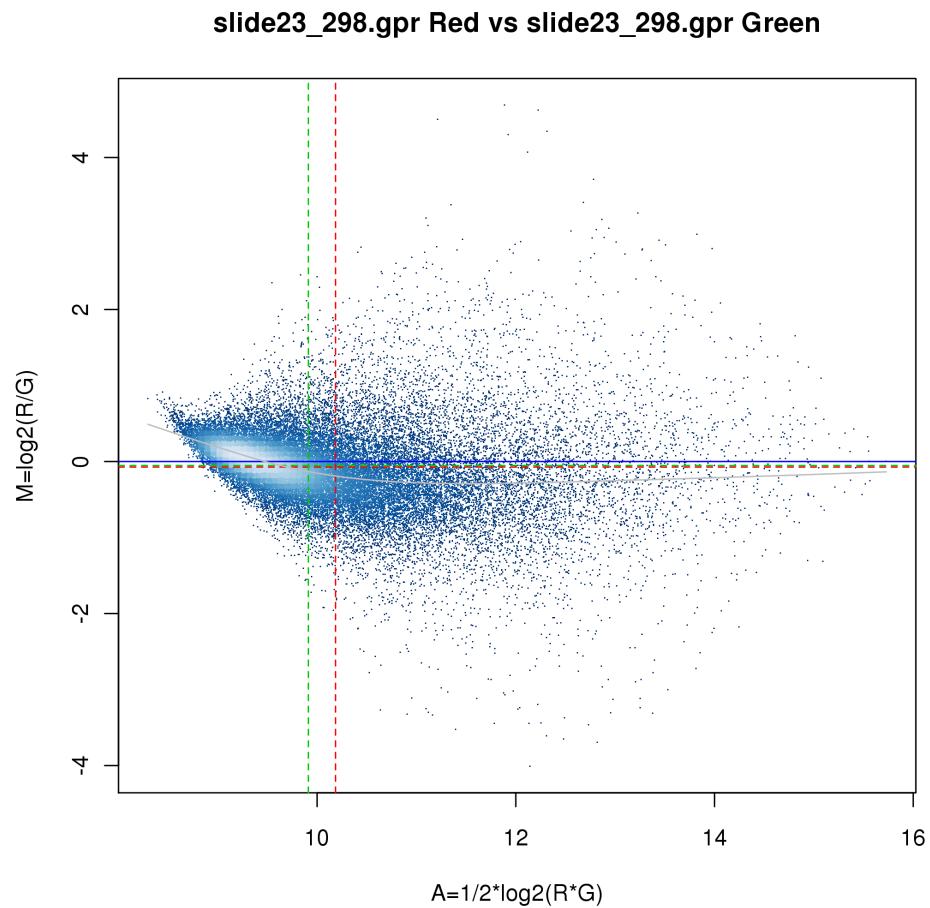


Figure 1.39: MA plot of array 15 (slide23_298.gpr). Raw data before background correction.

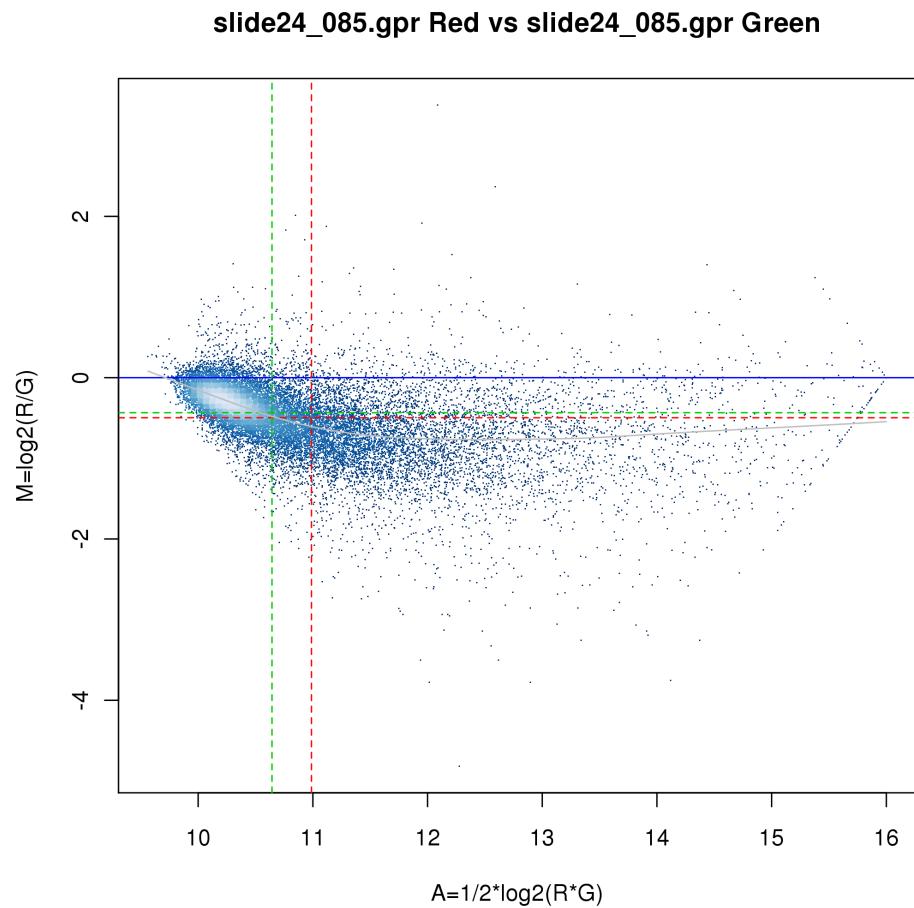


Figure 1.40: MA plot of array 16 (slide24_085.gpr). Raw data before background correction.

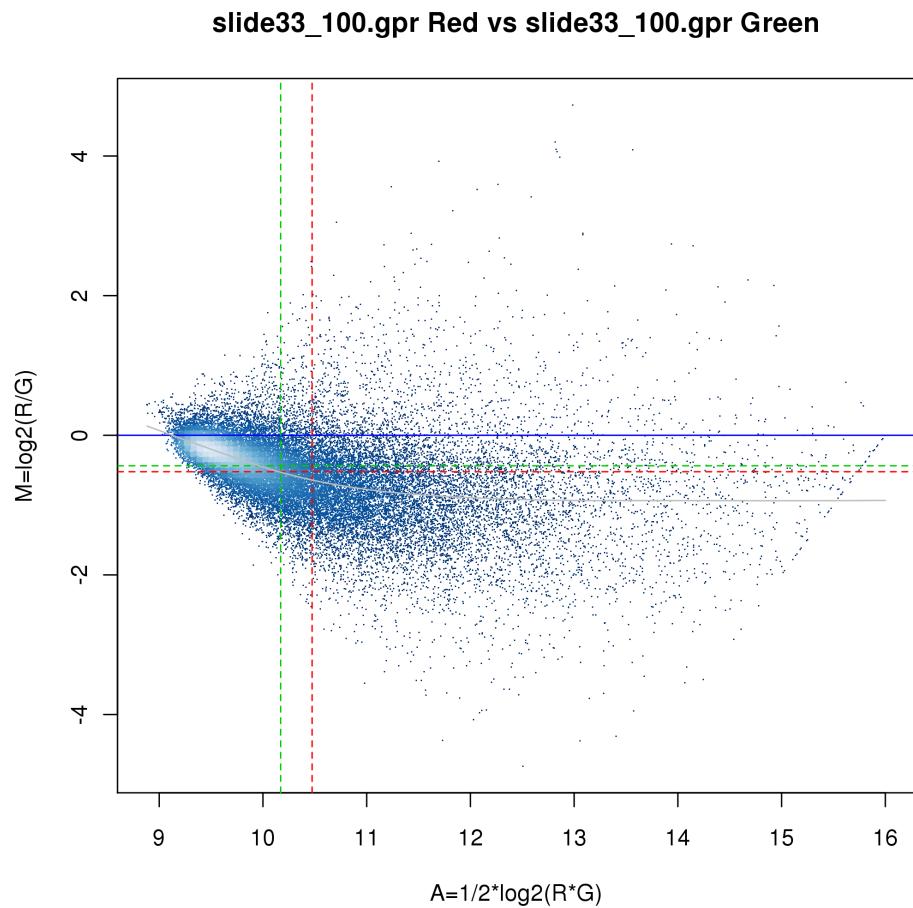


Figure 1.41: MA plot of array 17 (slide33_100.gpr). Raw data before background correction.

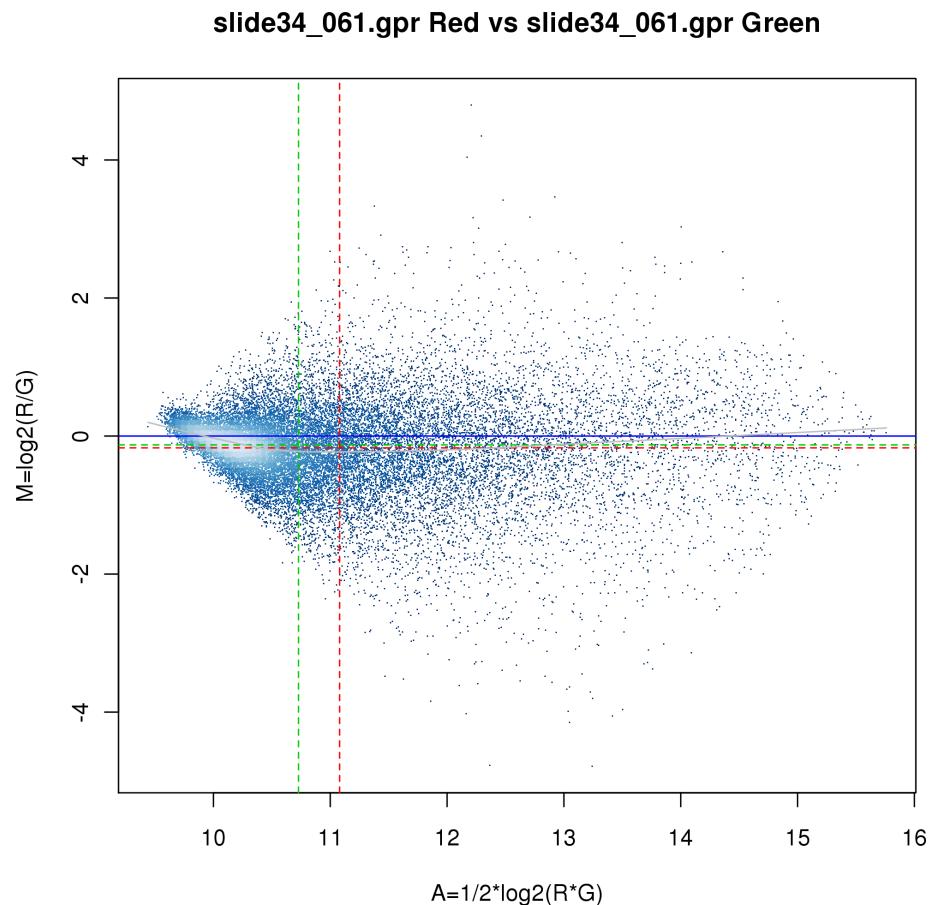


Figure 1.42: MA plot of array 18 (slide34_061.gpr). Raw data before background correction.

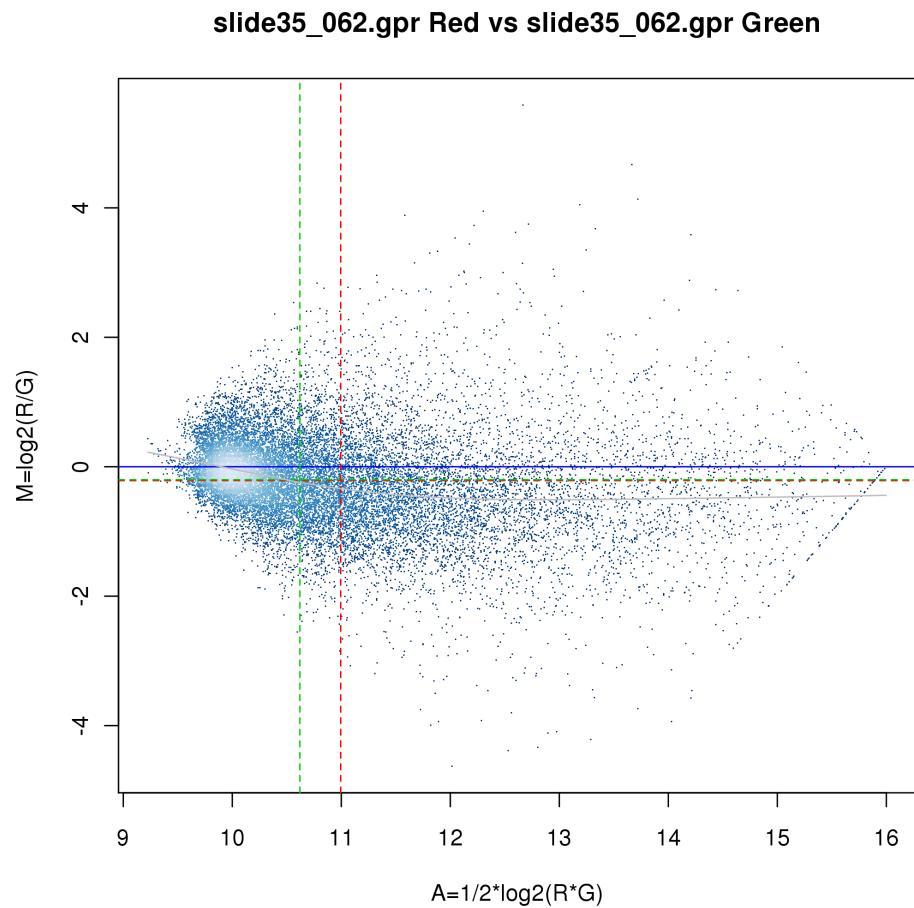


Figure 1.43: MA plot of array 19 (slide35_062.gpr). Raw data before background correction.

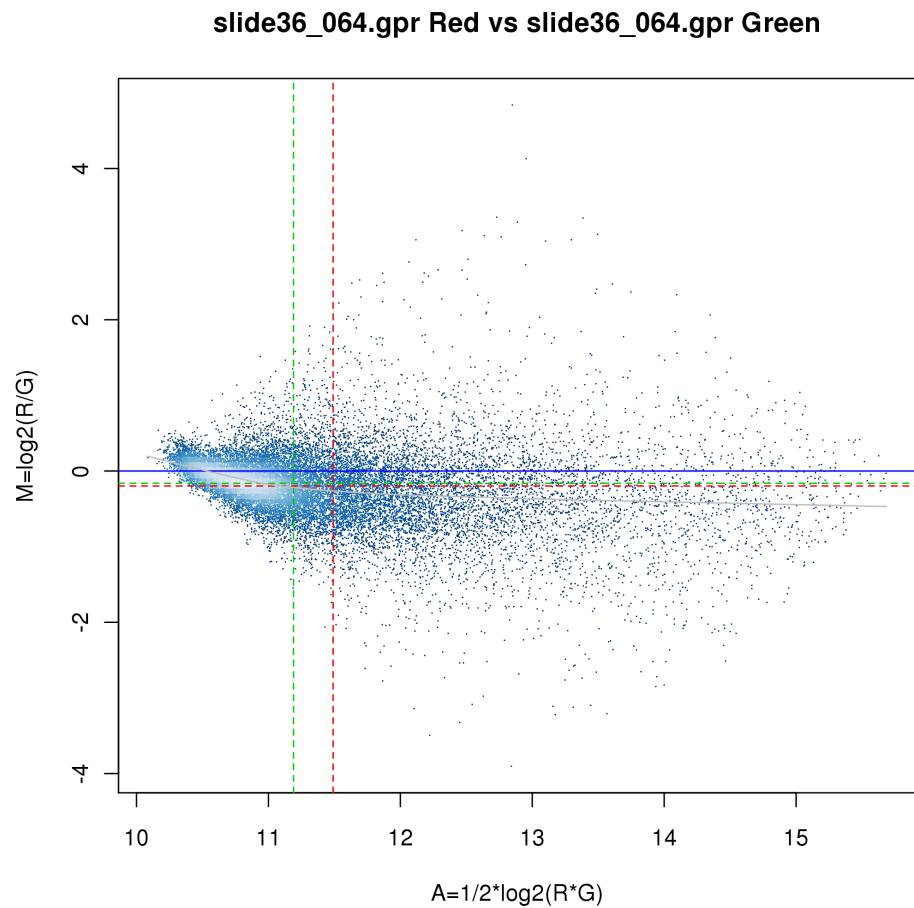


Figure 1.44: MA plot of array 20 (slide36_064.gpr). Raw data before background correction.

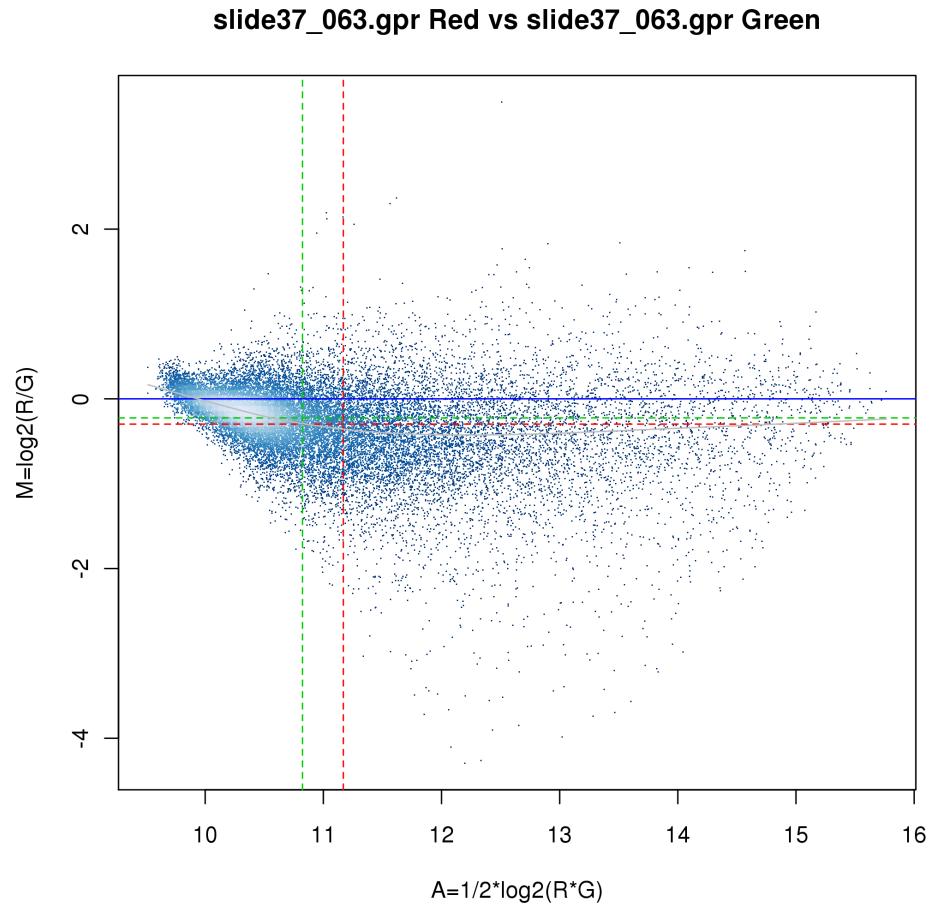


Figure 1.45: MA plot of array 21 (slide37_063.gpr). Raw data before background correction.

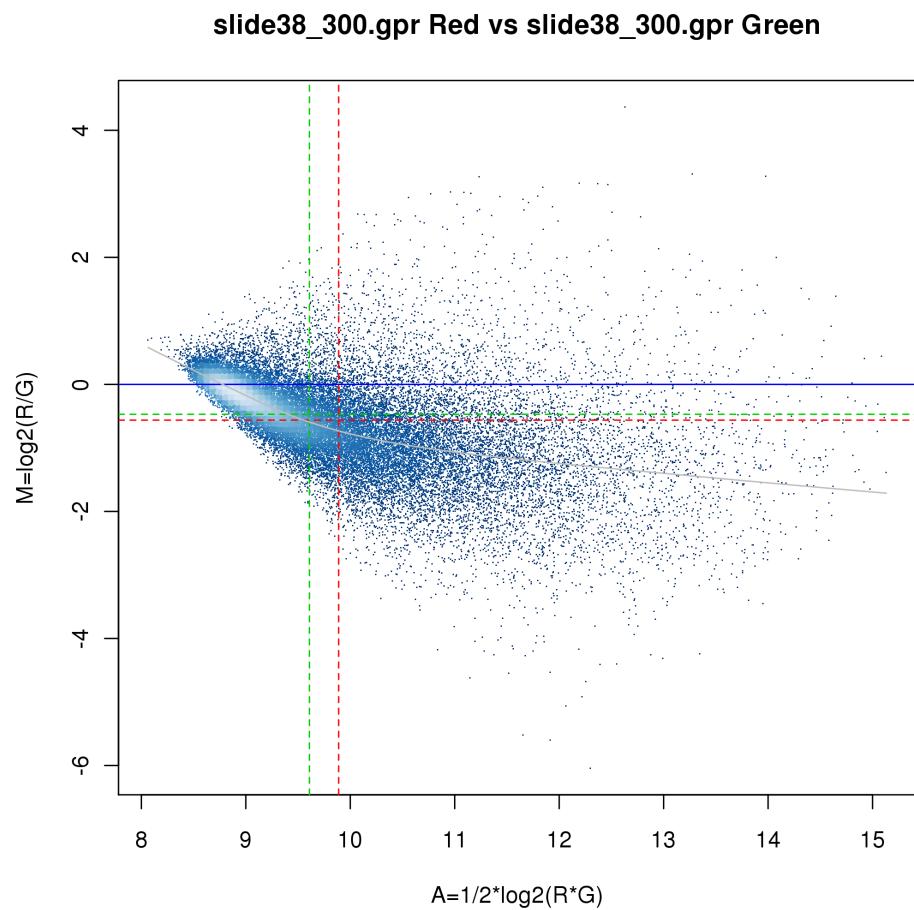


Figure 1.46: MA plot of array 22 (slide38_300.gpr). Raw data before background correction.

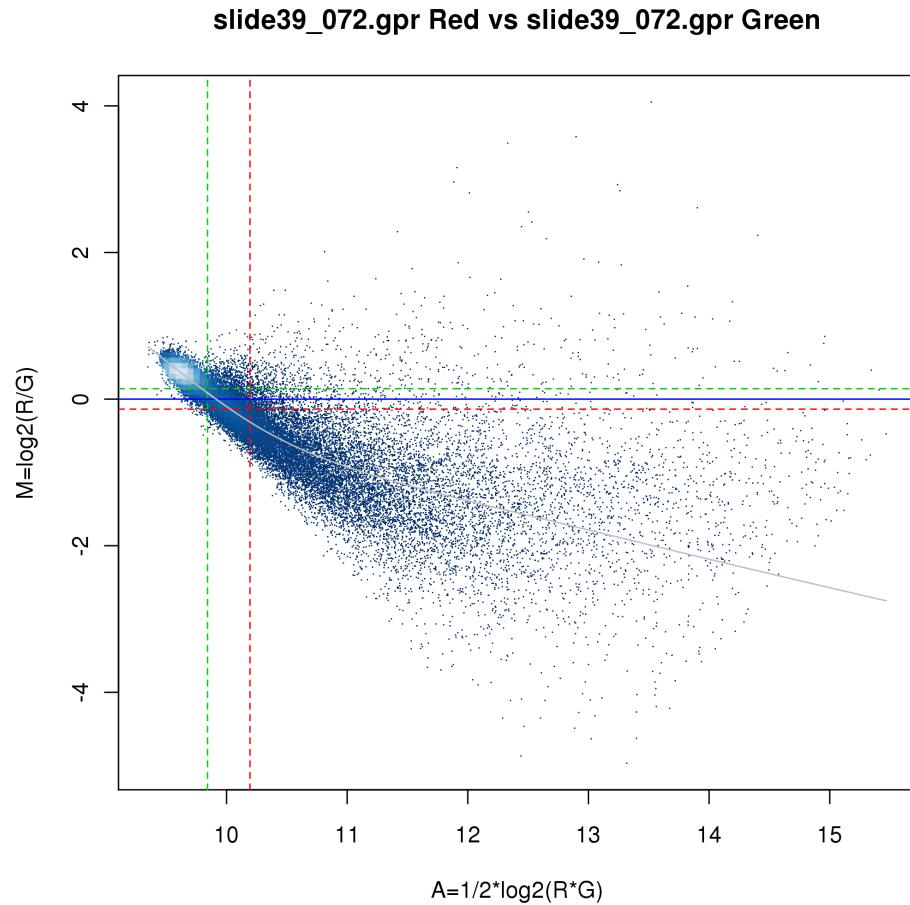


Figure 1.47: MA plot of array 23 (slide39_072.gpr). Raw data before background correction.

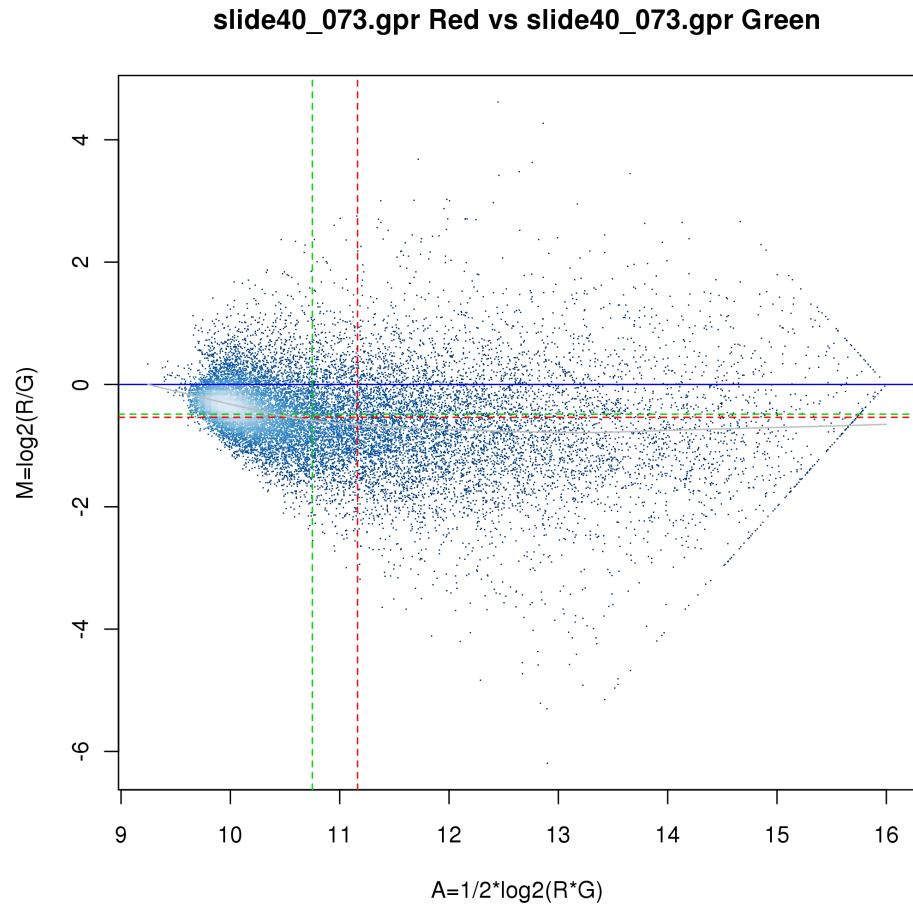


Figure 1.48: MA plot of array 24 (slide40_073.gpr). Raw data before background correction.

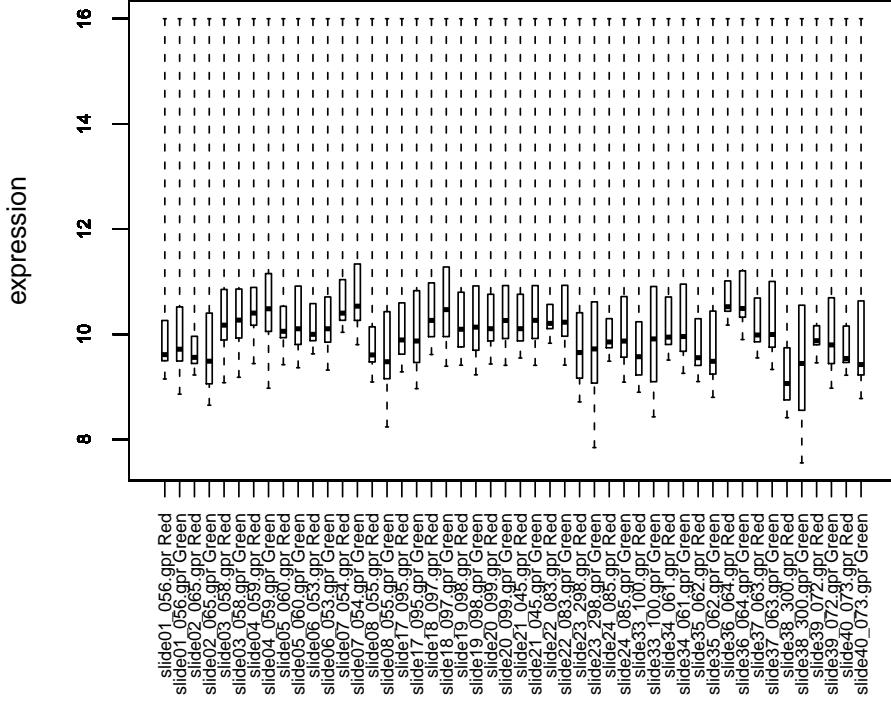


Figure 1.49: Boxplots of the signal intensities of each signal channel of the microarrays. Raw data before background correction.

```
> Slides.raw <- backgroundCorrect(Slides.raw, method = "minimum")
```

Next diagnostic plots of the background corrected raw data will be drawn.

```
> Dummy <- newMadbSet(Slides.raw)
```

```
Converting a limma RGLList into a MadbSet...
```

```
Setting the weights... a weights of 0 means the gene was flagged, a weights of one means the signal is ok!
```

```
Inserting available annotation into the slot @genes
```

```
Inserting available annotation into the slot @genes
```

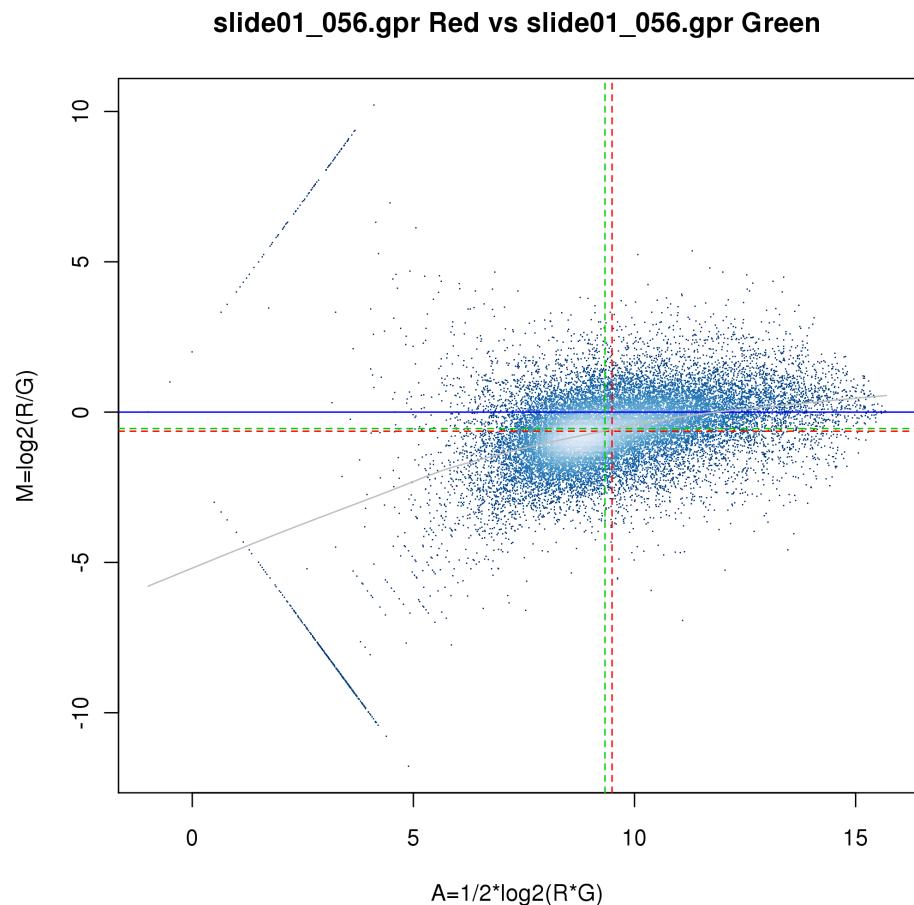


Figure 1.50: MA plot of array 1 (slide01_056.gpr). Raw data after background correction.

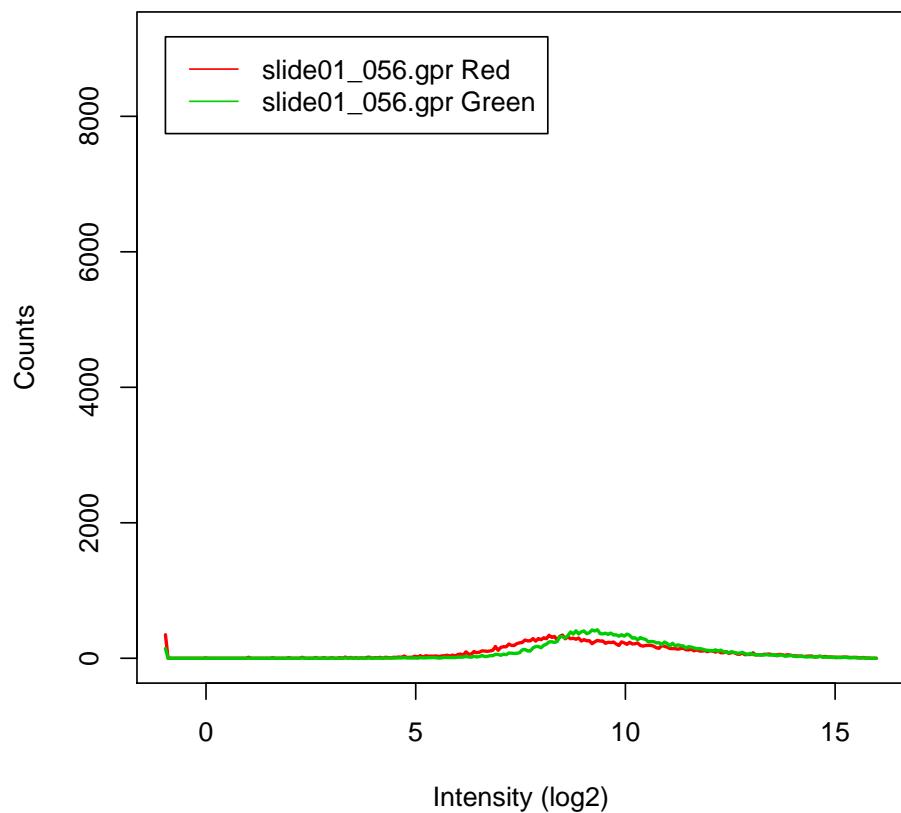


Figure 1.51: Histogram of the array 1 (slide01_056.gpr). Raw data after background correction.

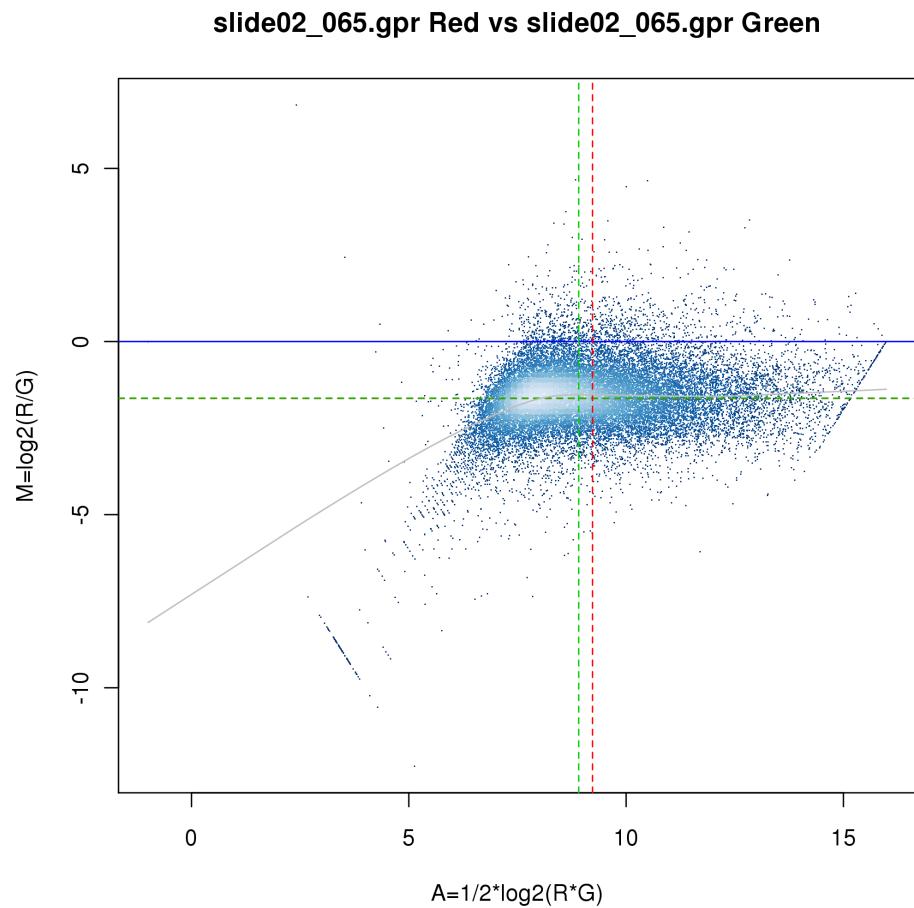


Figure 1.52: MA plot of array 2 (slide02_065.gpr). Raw data after background correction.

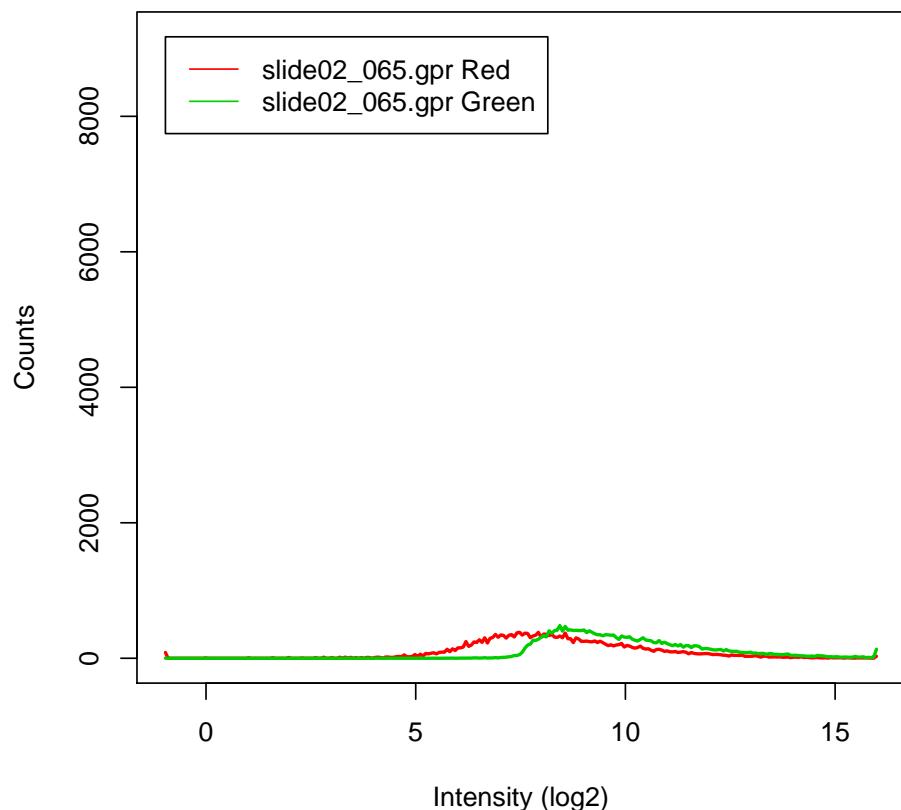


Figure 1.53: Histogram of the array 2 (slide02_065.gpr). Raw data after background correction.

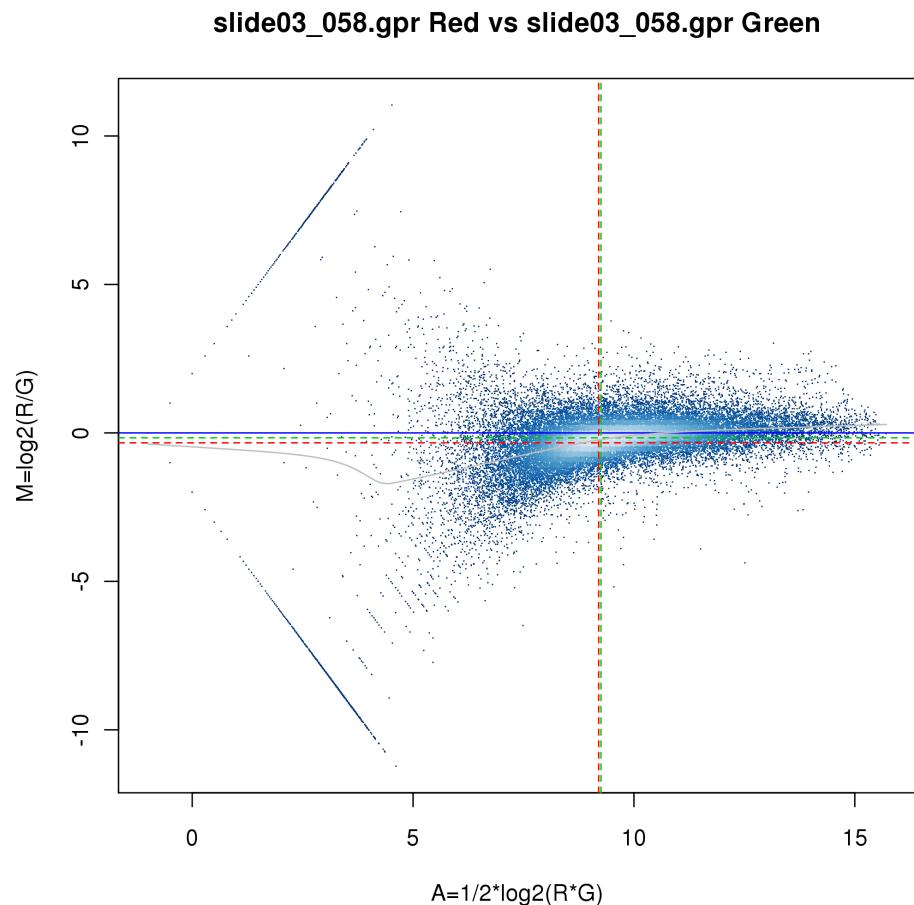


Figure 1.54: MA plot of array 3 (slide03_058.gpr). Raw data after background correction.

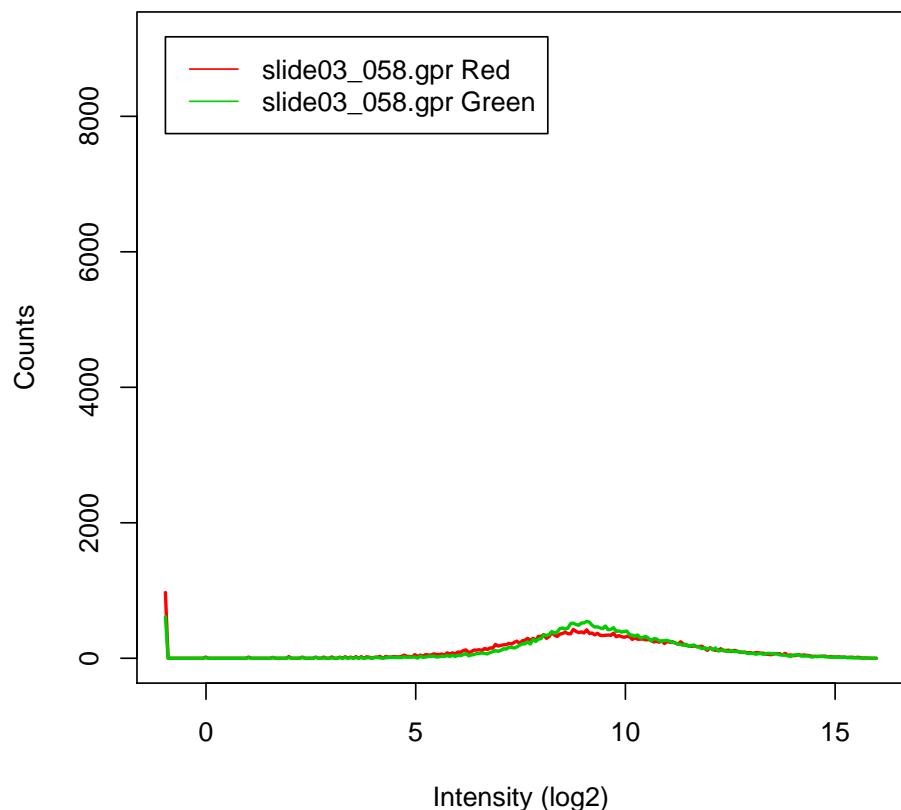


Figure 1.55: Histogram of the array 3 (slide03_058.gpr). Raw data after background correction.

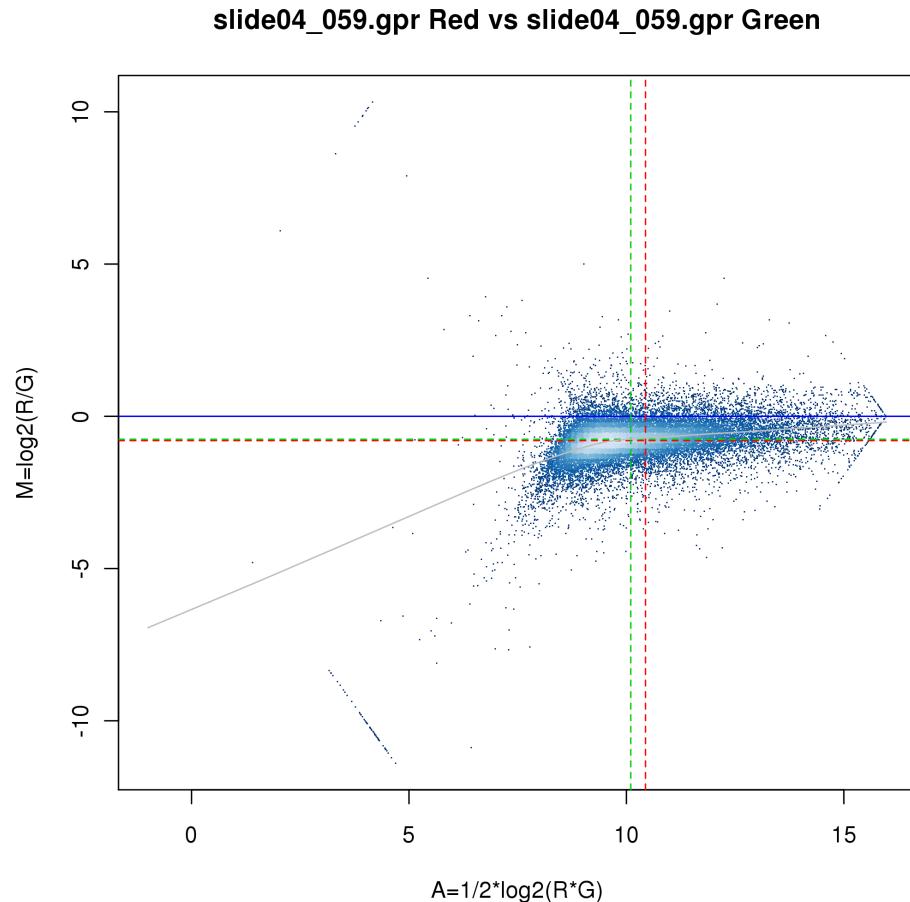


Figure 1.56: MA plot of array 4 (slide04_059.gpr). Raw data after background correction.

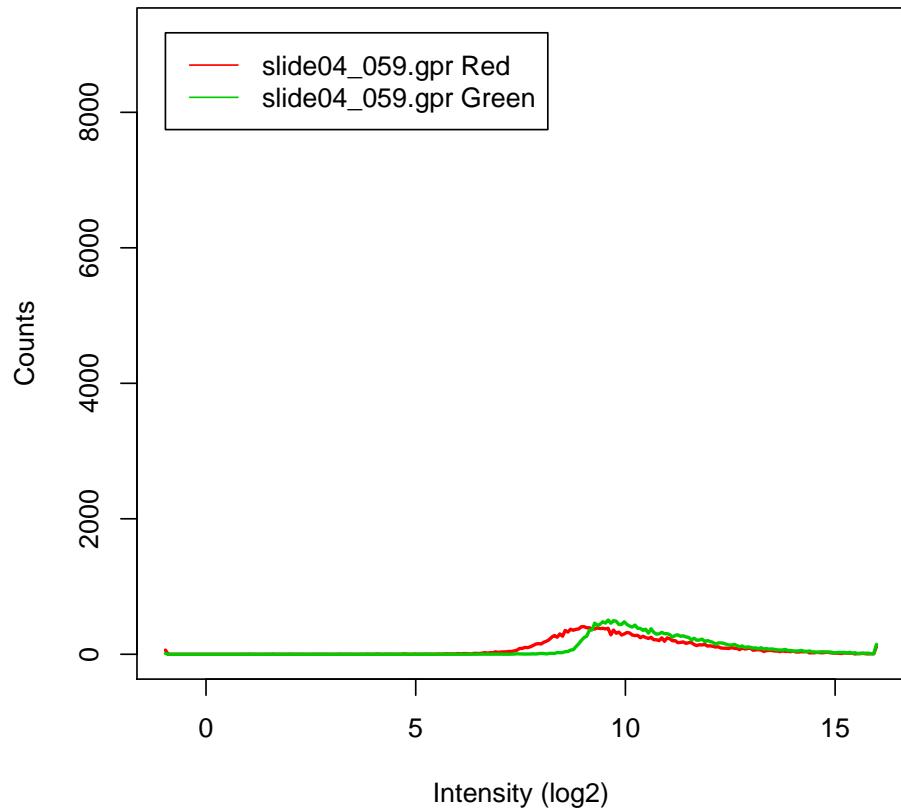


Figure 1.57: Histogram of the array 4 (slide04_059.gpr). Raw data after background correction.

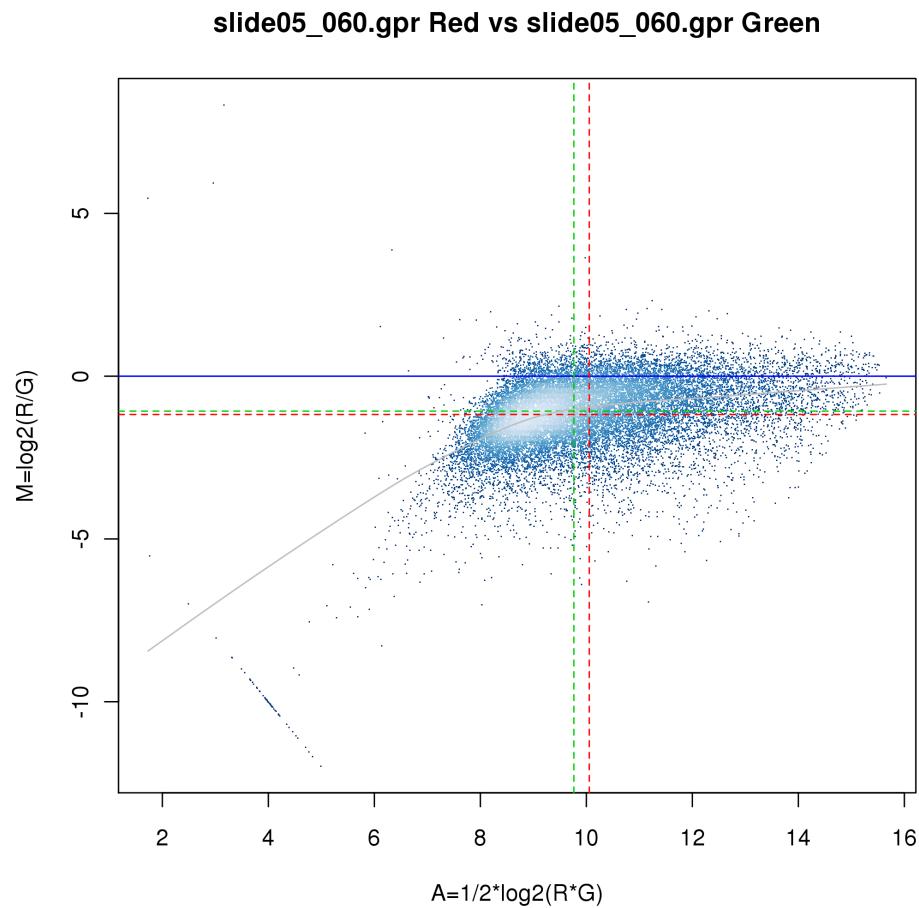


Figure 1.58: MA plot of array 5 (slide05_060.gpr). Raw data after background correction.

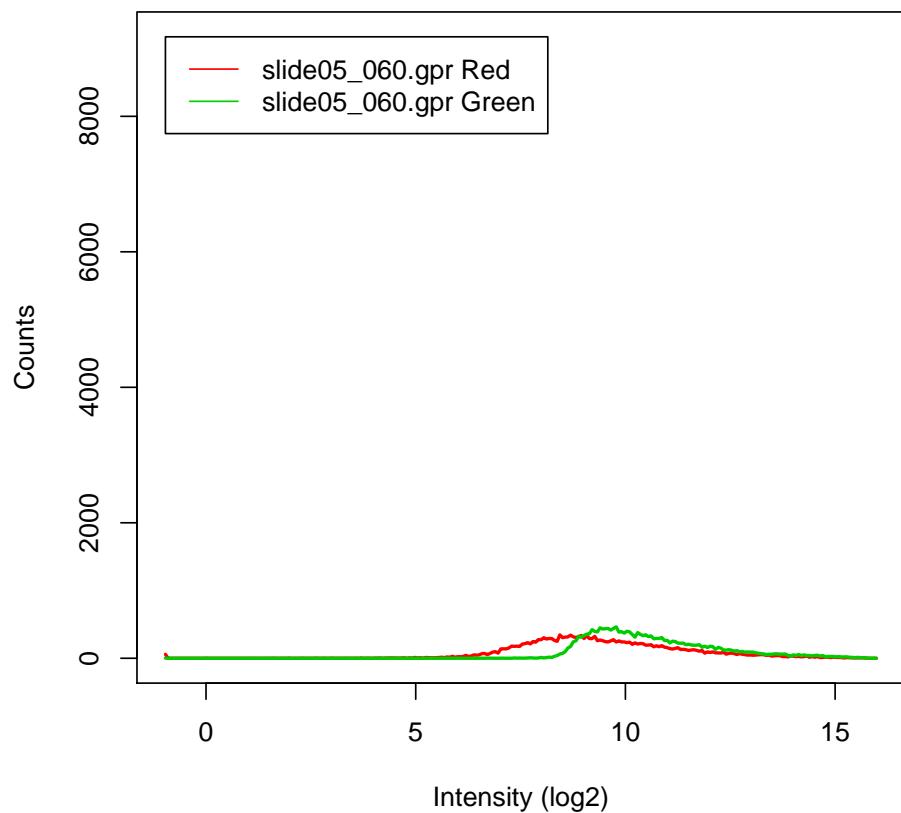


Figure 1.59: Histogram of the array 5 (slide05_060.gpr). Raw data after background correction.

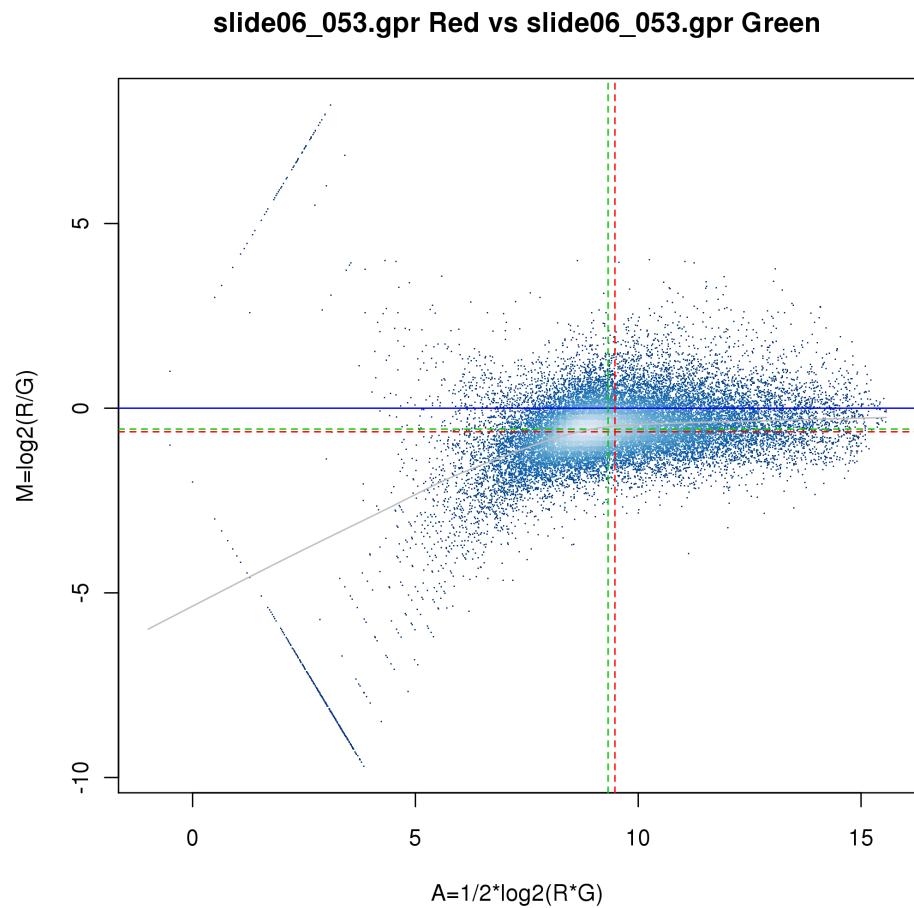


Figure 1.60: MA plot of array 6 (slide06_053.gpr). Raw data after background correction.

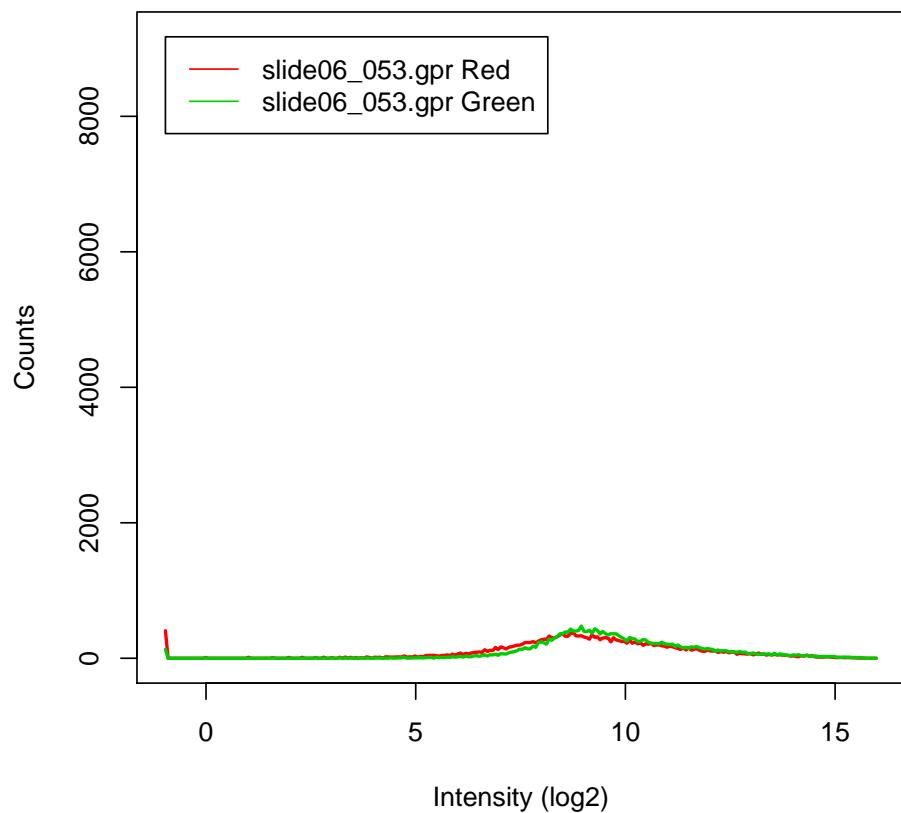


Figure 1.61: Histogram of the array 6 (slide06_053.gpr). Raw data after background correction.

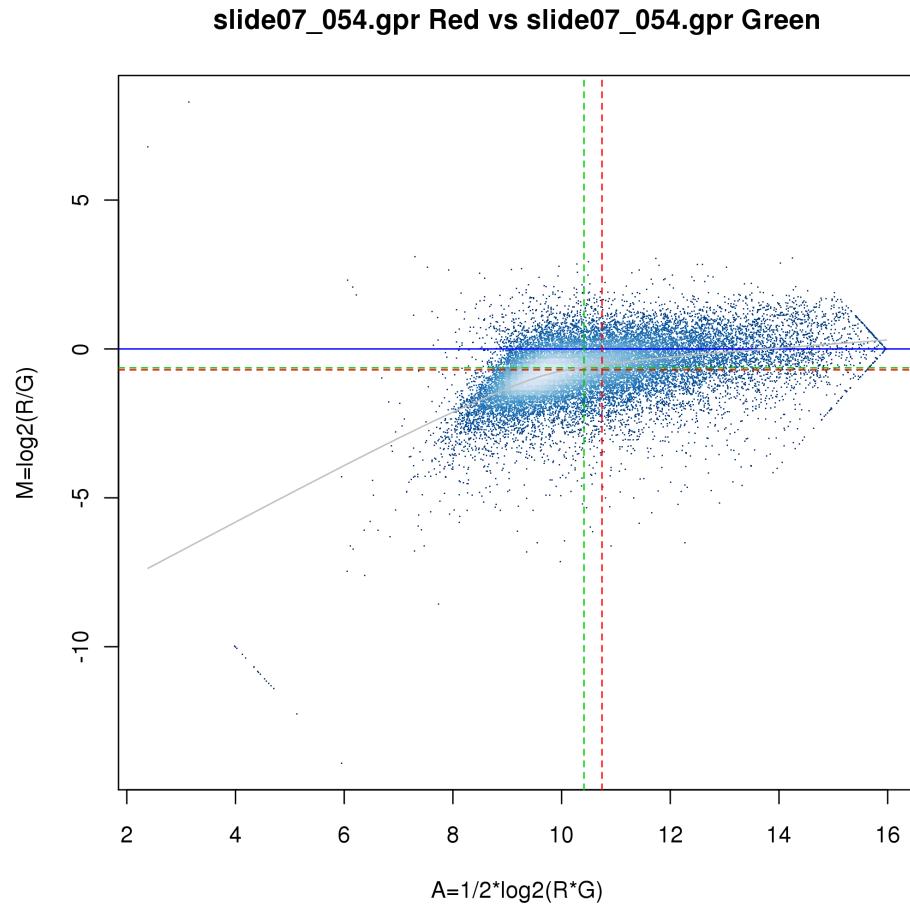


Figure 1.62: MA plot of array 7 (slide07_054.gpr). Raw data after background correction.

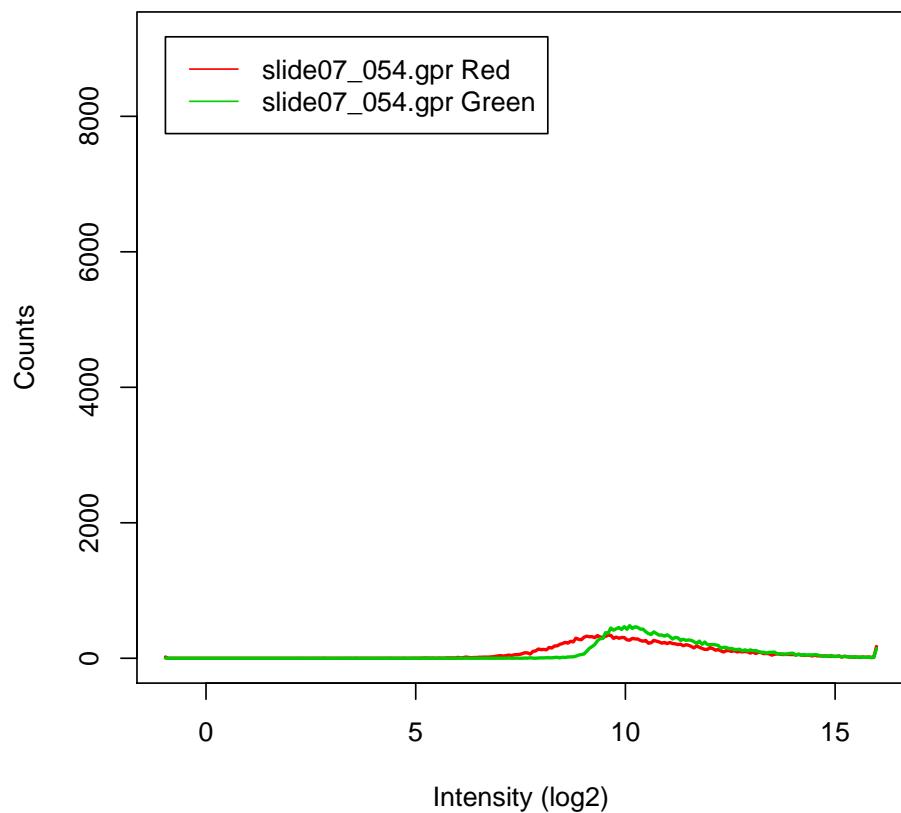


Figure 1.63: Histogram of the array 7 (slide07_054.gpr). Raw data after background correction.

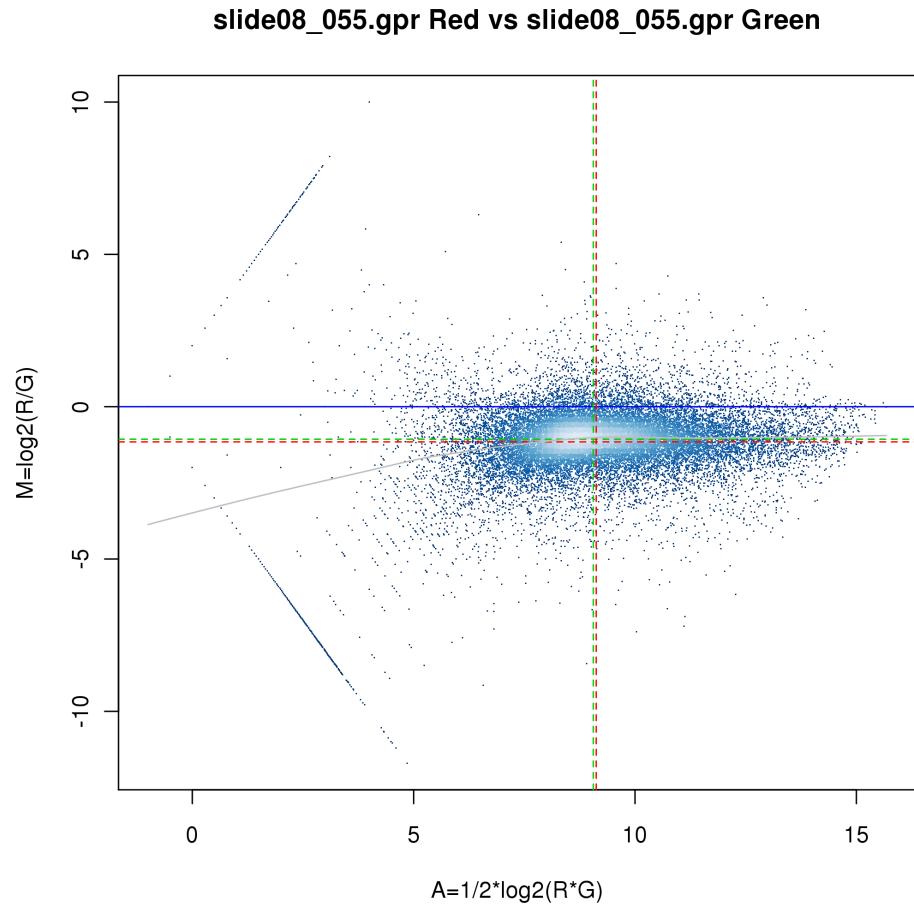


Figure 1.64: MA plot of array 8 (slide08_055.gpr). Raw data after background correction.

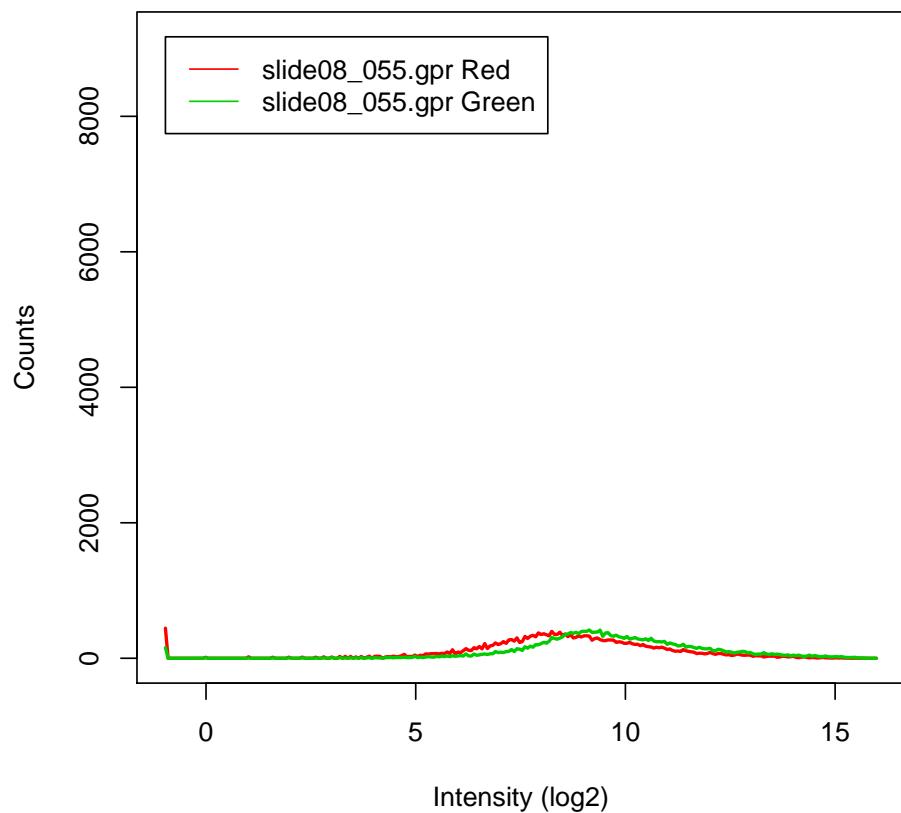


Figure 1.65: Histogram of the array 8 (slide08_055.gpr). Raw data after background correction.

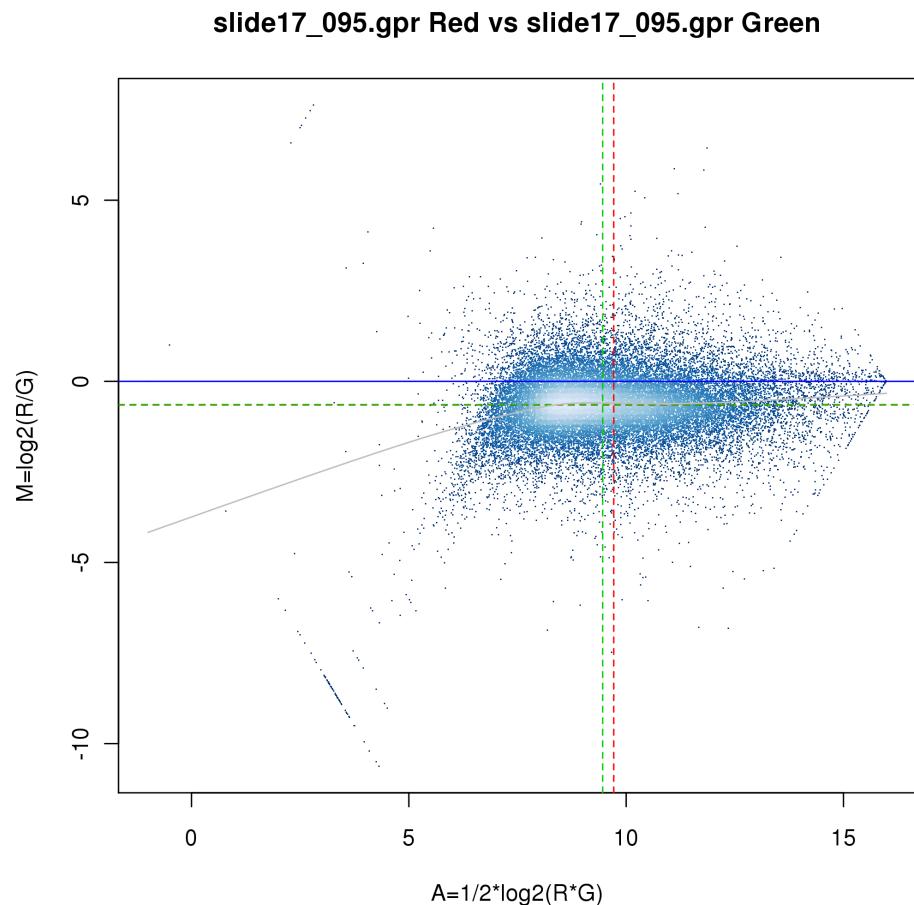


Figure 1.66: MA plot of array 9 (slide17_095.gpr). Raw data after background correction.

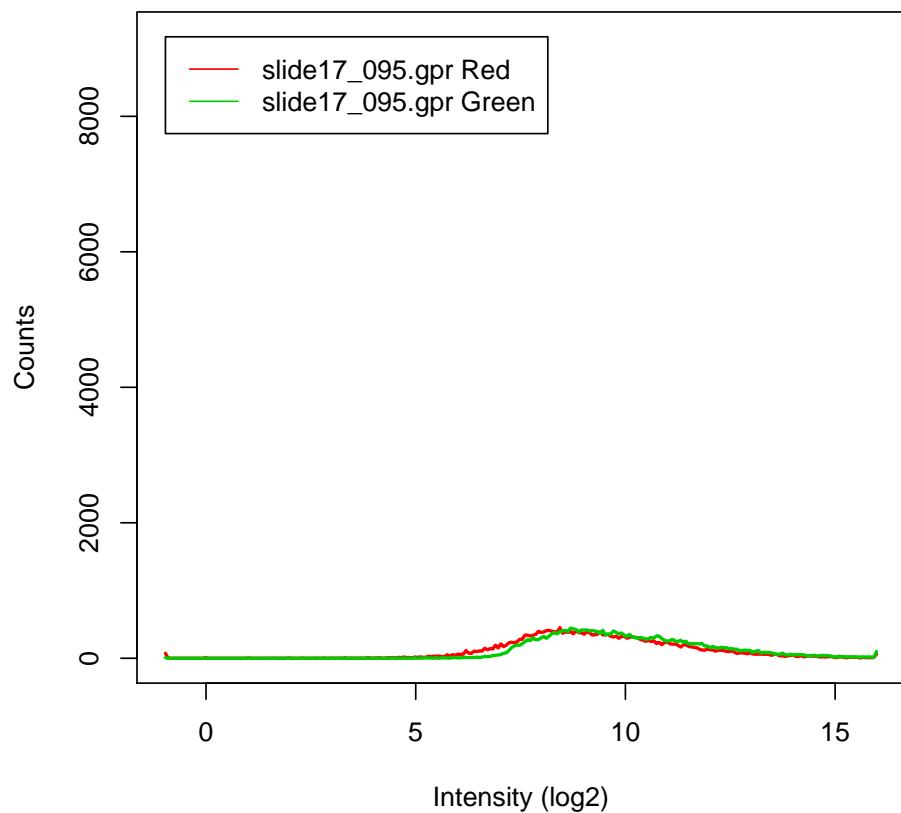


Figure 1.67: Histogram of the array 9 (slide17_095.gpr). Raw data after background correction.

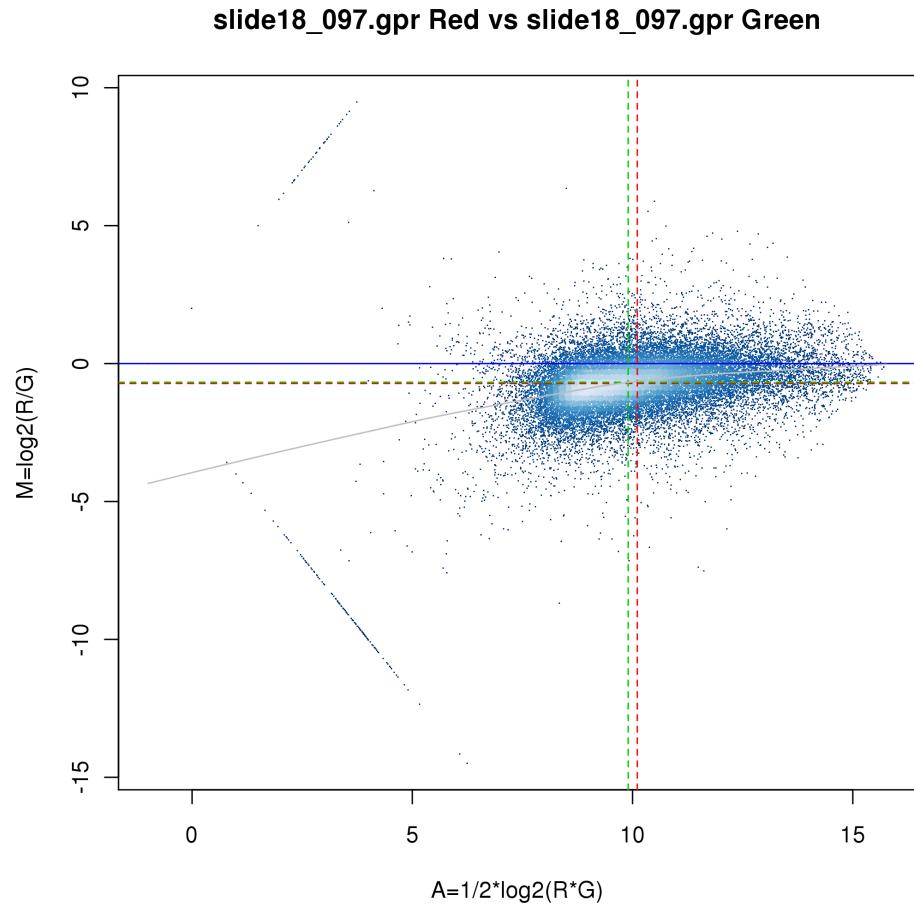


Figure 1.68: MA plot of array 10 (slide18_097.gpr). Raw data after background correction.

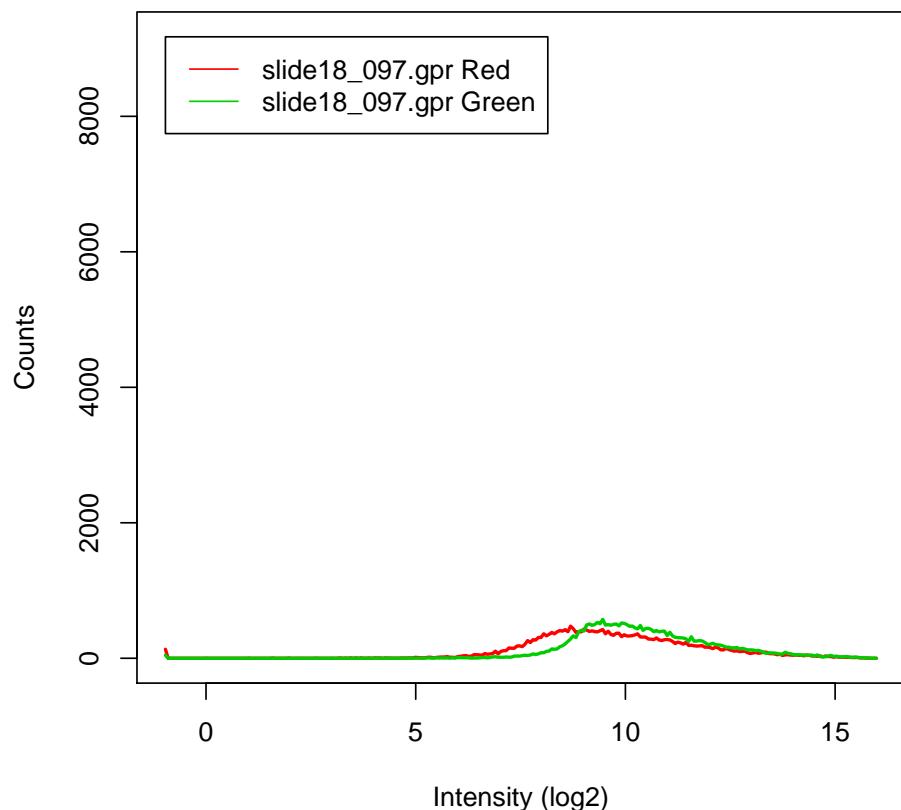


Figure 1.69: Histogram of the array 10 (slide18_097.gpr). Raw data after background correction.

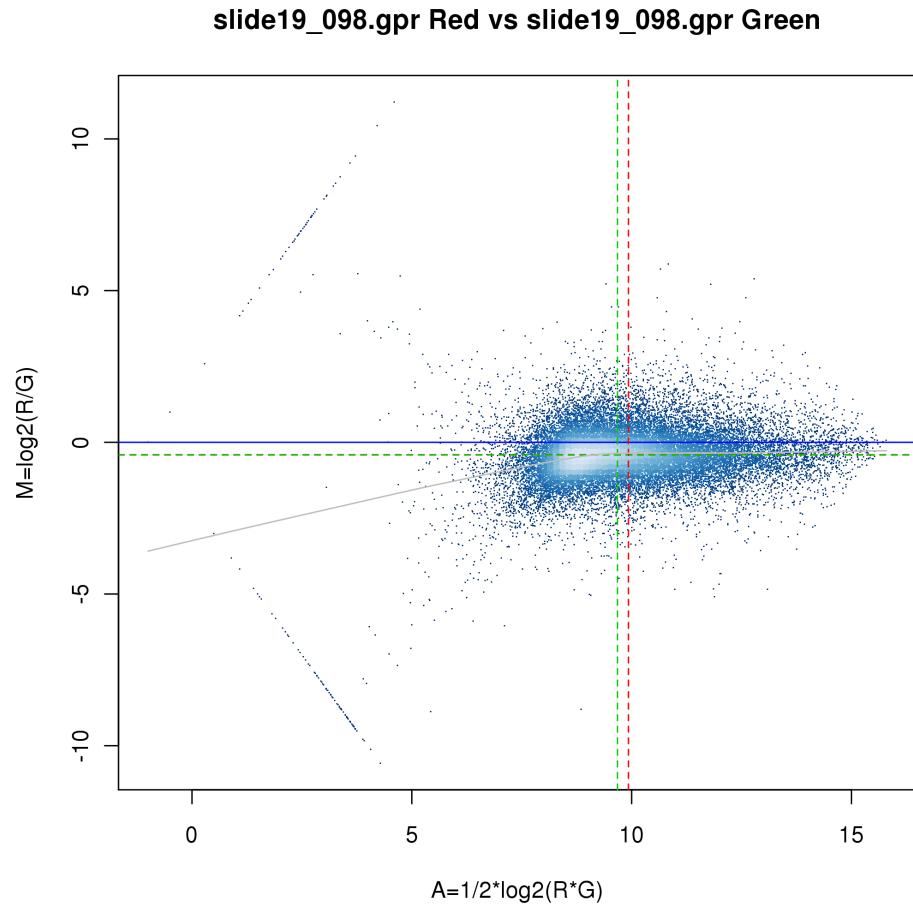


Figure 1.70: MA plot of array 11 (slide19_098.gpr). Raw data after background correction.

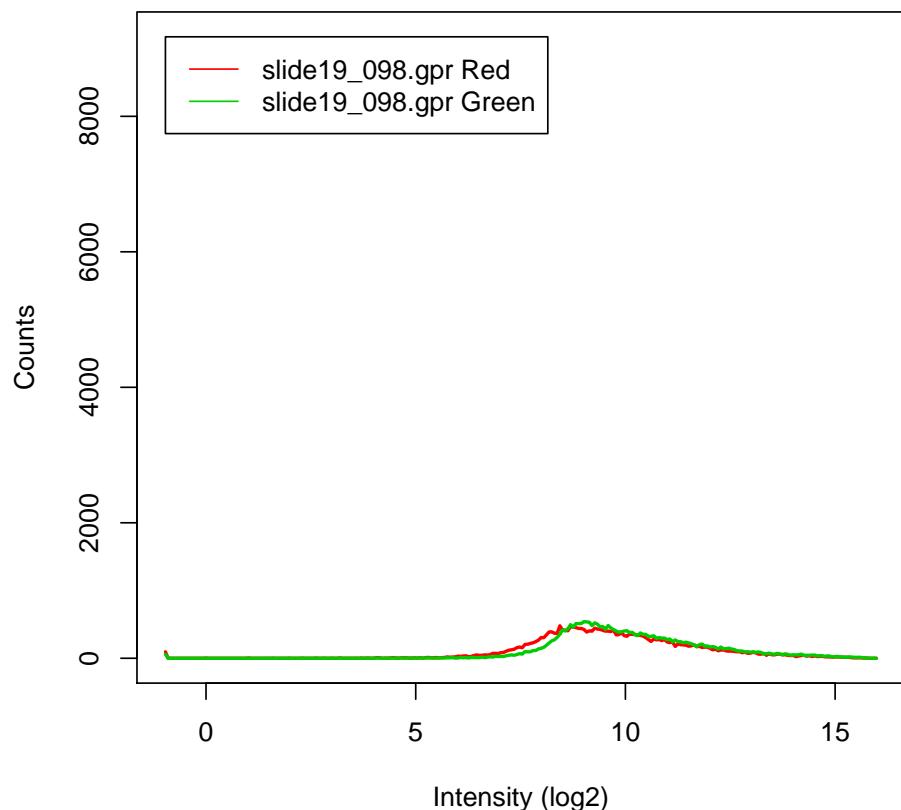


Figure 1.71: Histogram of the array 11 (slide19_098.gpr). Raw data after background correction.

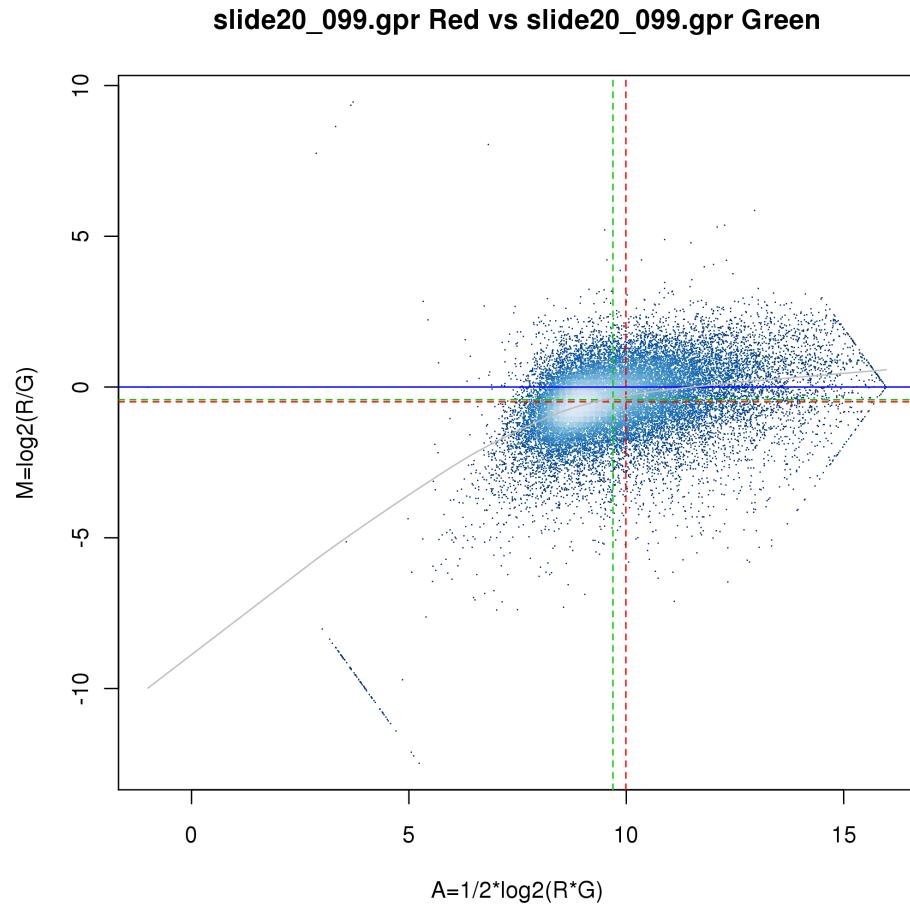


Figure 1.72: MA plot of array 12 (slide20_099.gpr). Raw data after background correction.

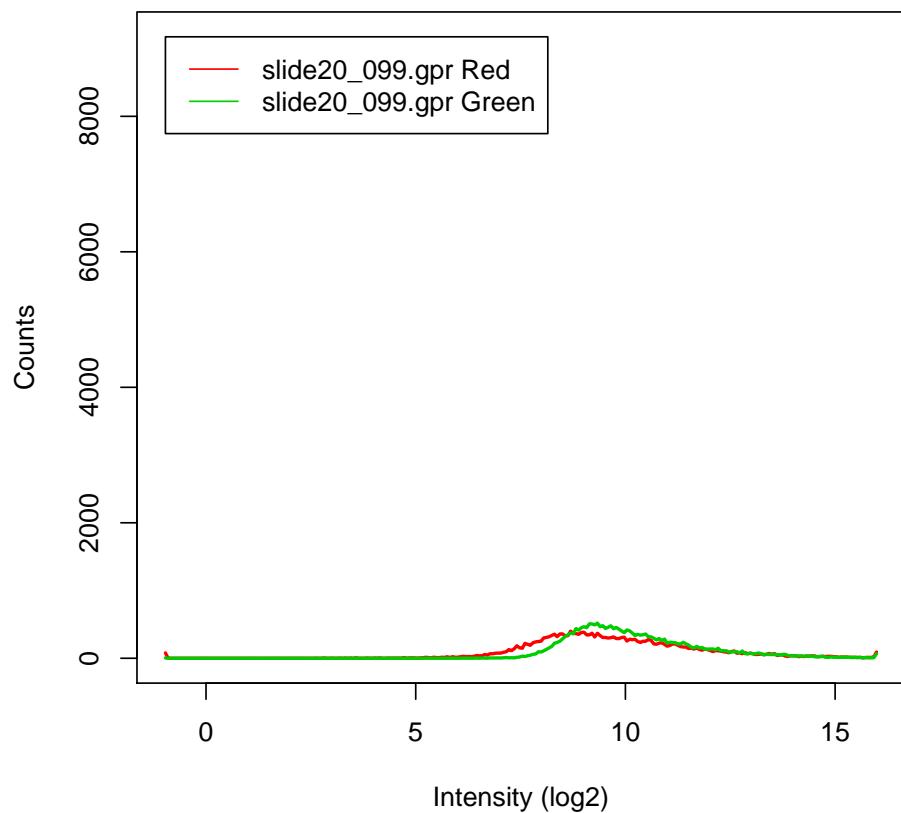


Figure 1.73: Histogram of the array 12 (slide20_099.gpr). Raw data after background correction.

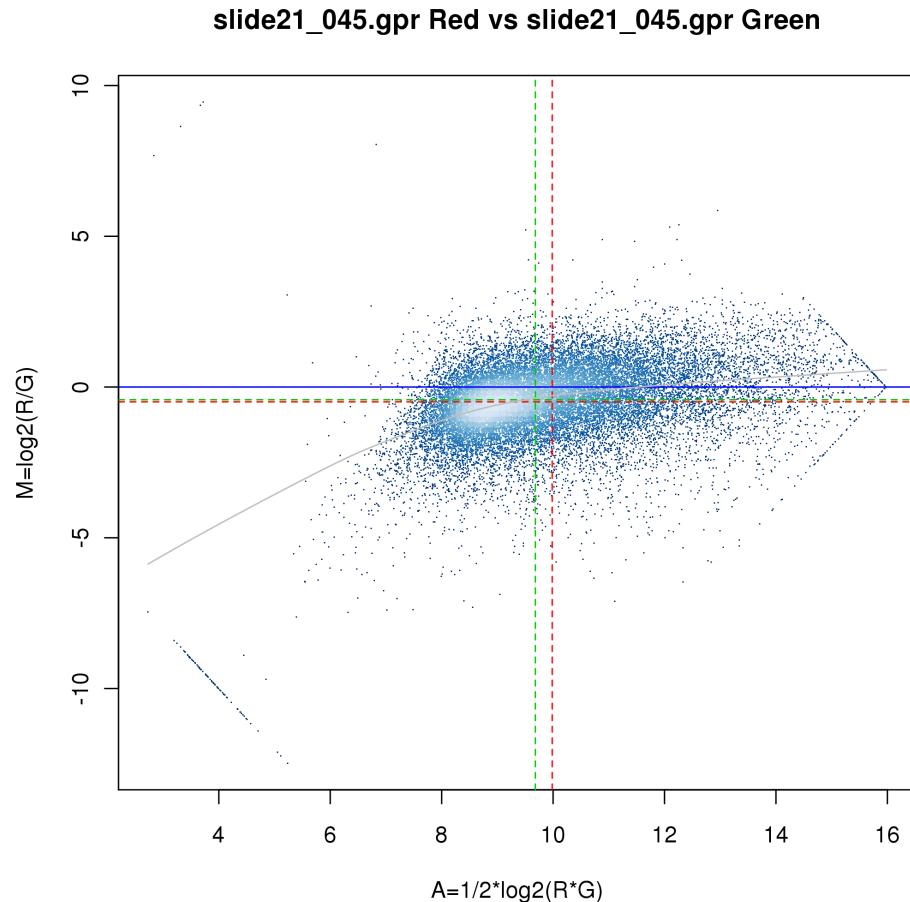


Figure 1.74: MA plot of array 13 (slide21_045.gpr). Raw data after background correction.

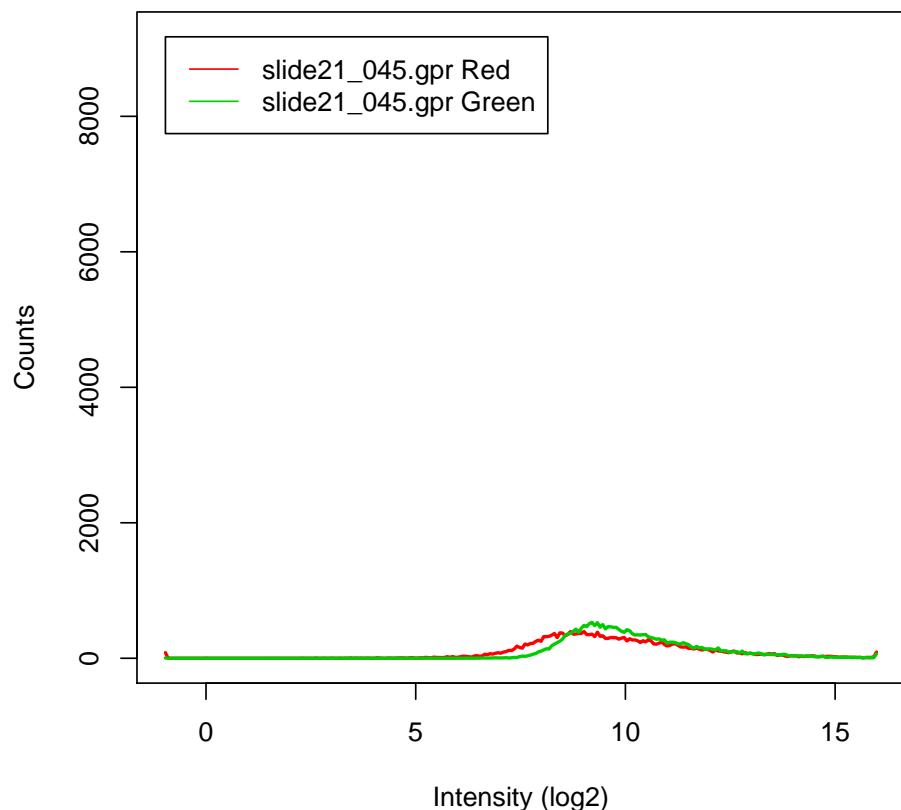


Figure 1.75: Histogram of the array 13 (slide21_045.gpr). Raw data after background correction.

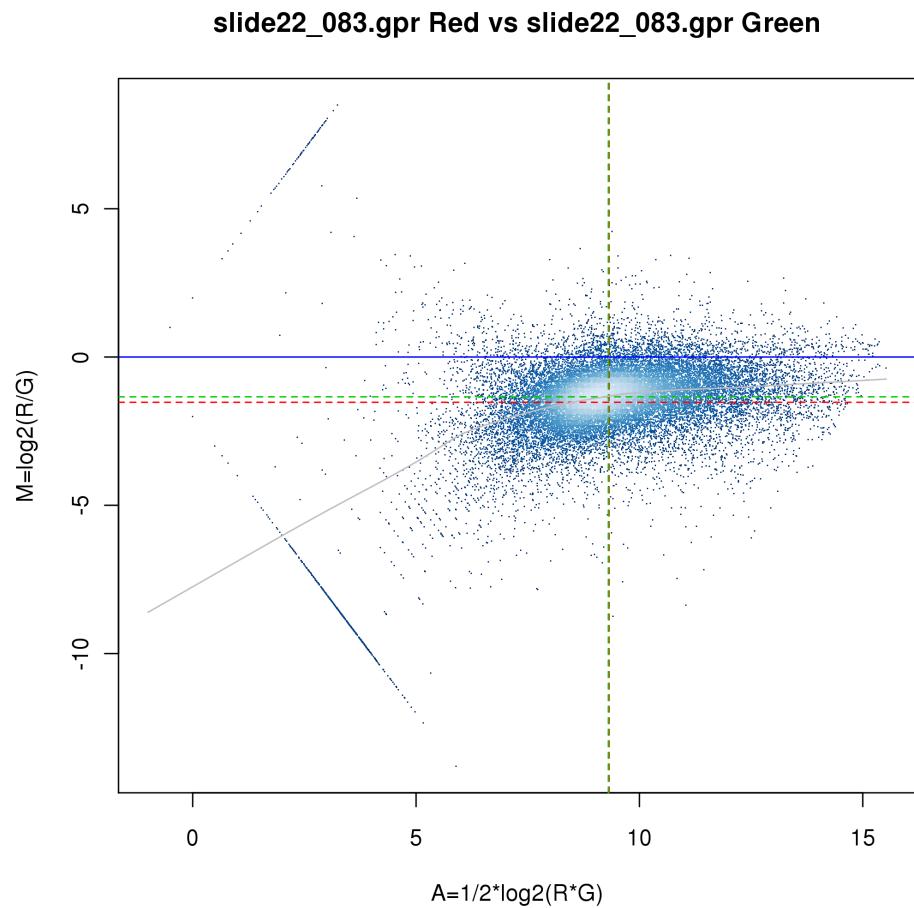


Figure 1.76: MA plot of array 14 (slide22_083.gpr). Raw data after background correction.

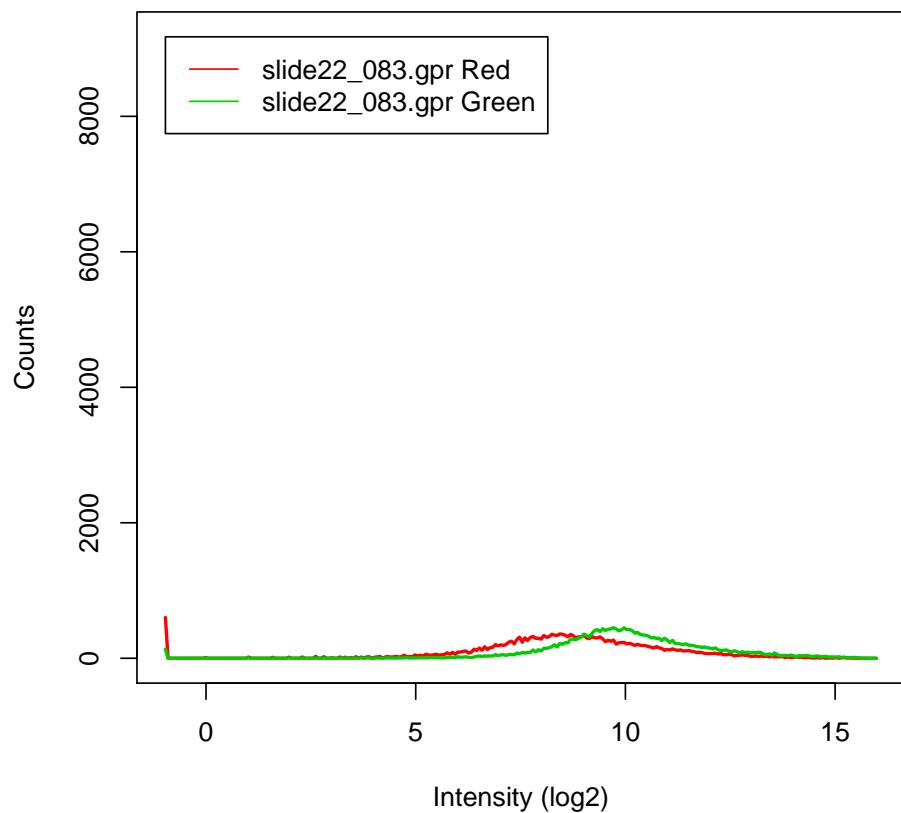


Figure 1.77: Histogram of the array 14 (slide22_083.gpr). Raw data after background correction.

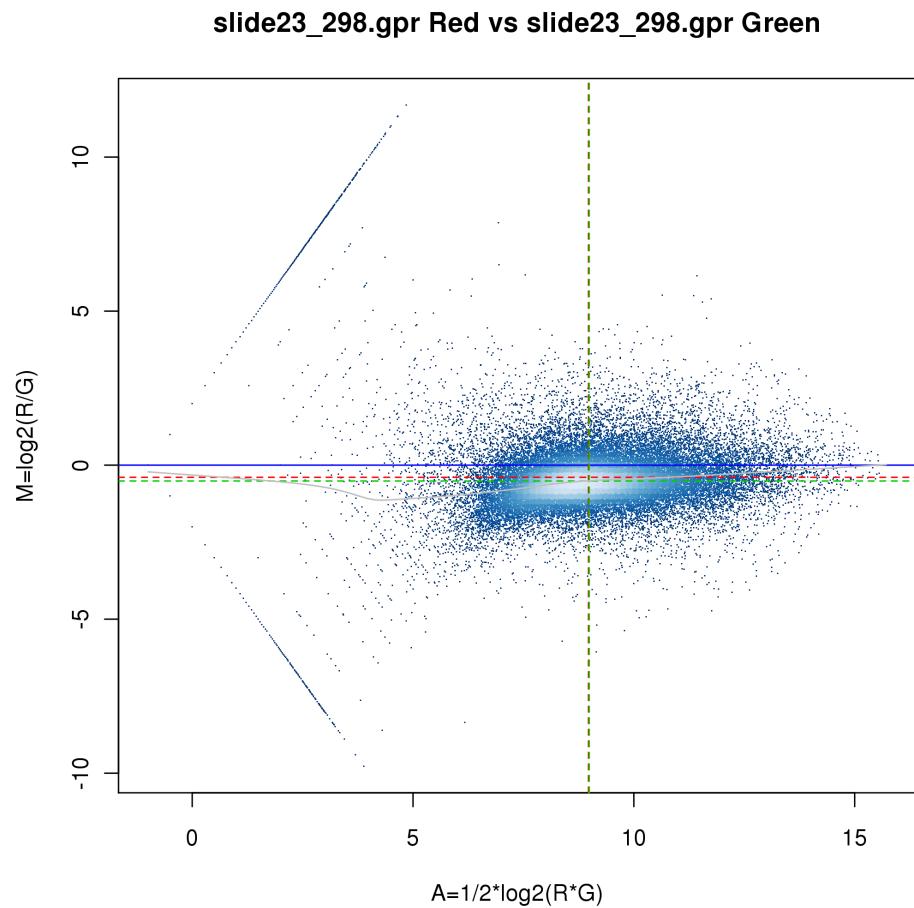


Figure 1.78: MA plot of array 15 (slide23_298.gpr). Raw data after background correction.

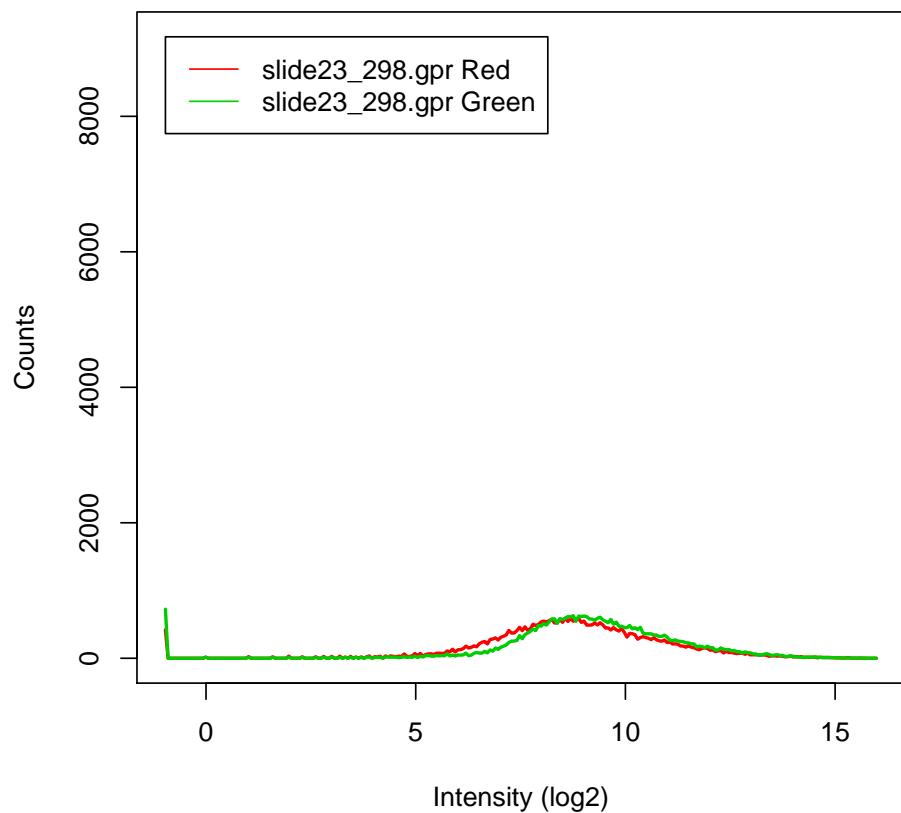


Figure 1.79: Histogram of the array 15 (slide23_298.gpr). Raw data after background correction.

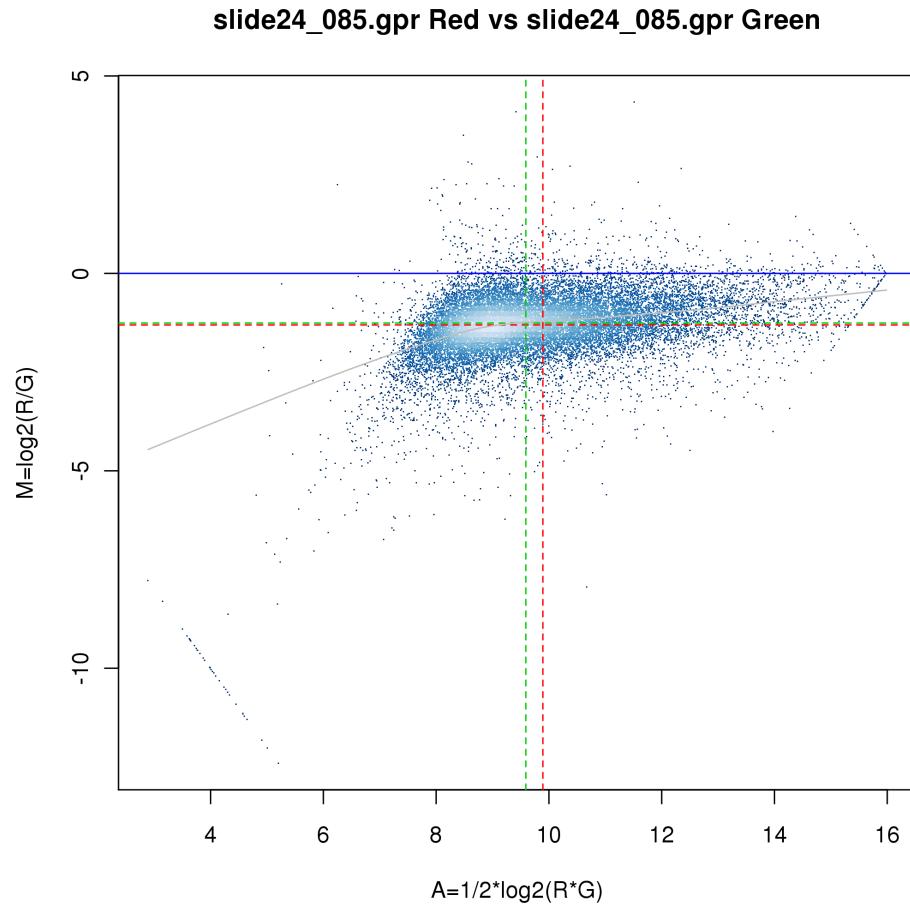


Figure 1.80: MA plot of array 16 (slide24_085.gpr). Raw data after background correction.

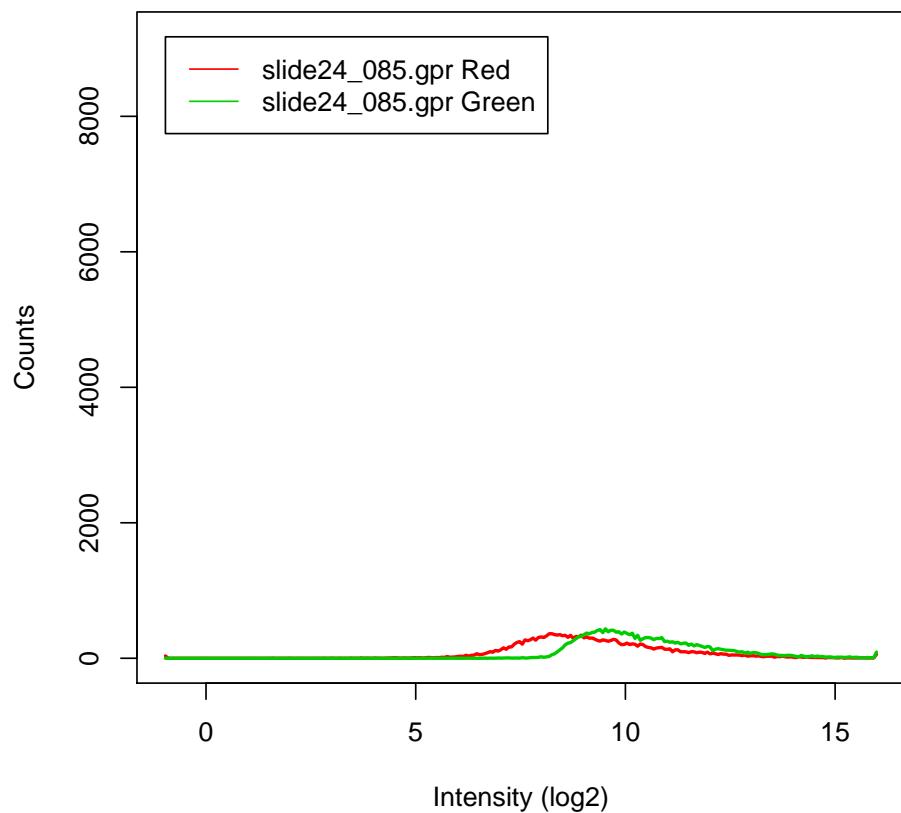


Figure 1.81: Histogram of the array 16 (slide24_085.gpr). Raw data after background correction.

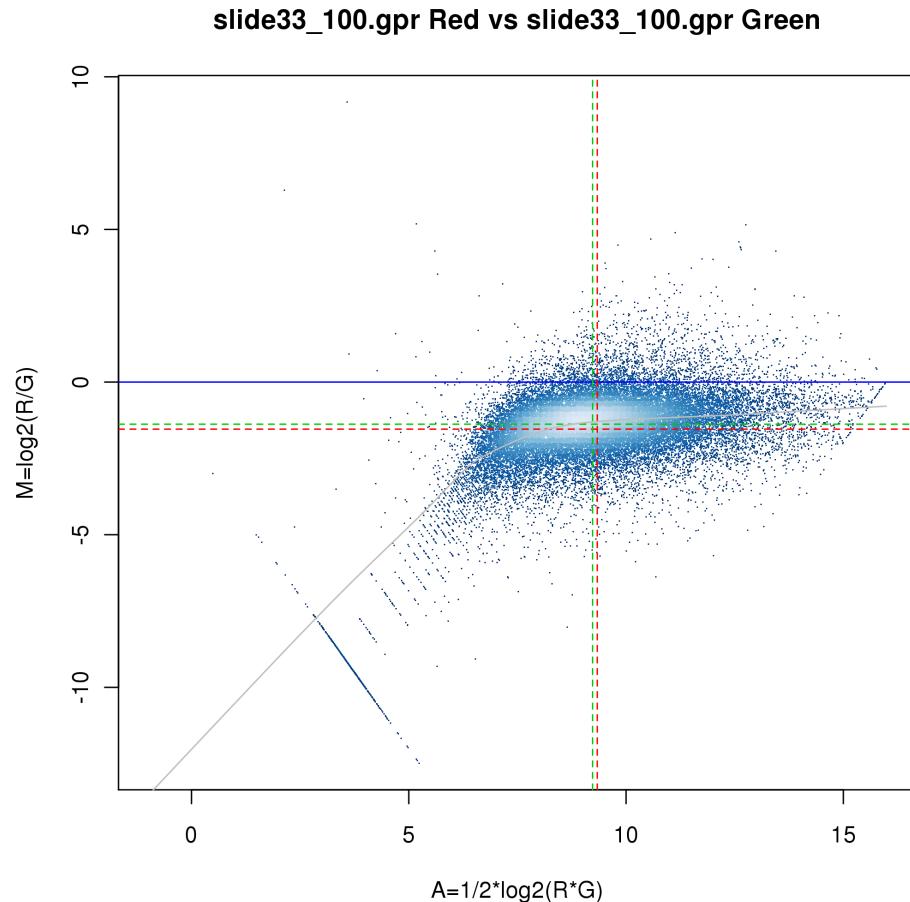


Figure 1.82: MA plot of array 17 (slide33_100.gpr). Raw data after background correction.

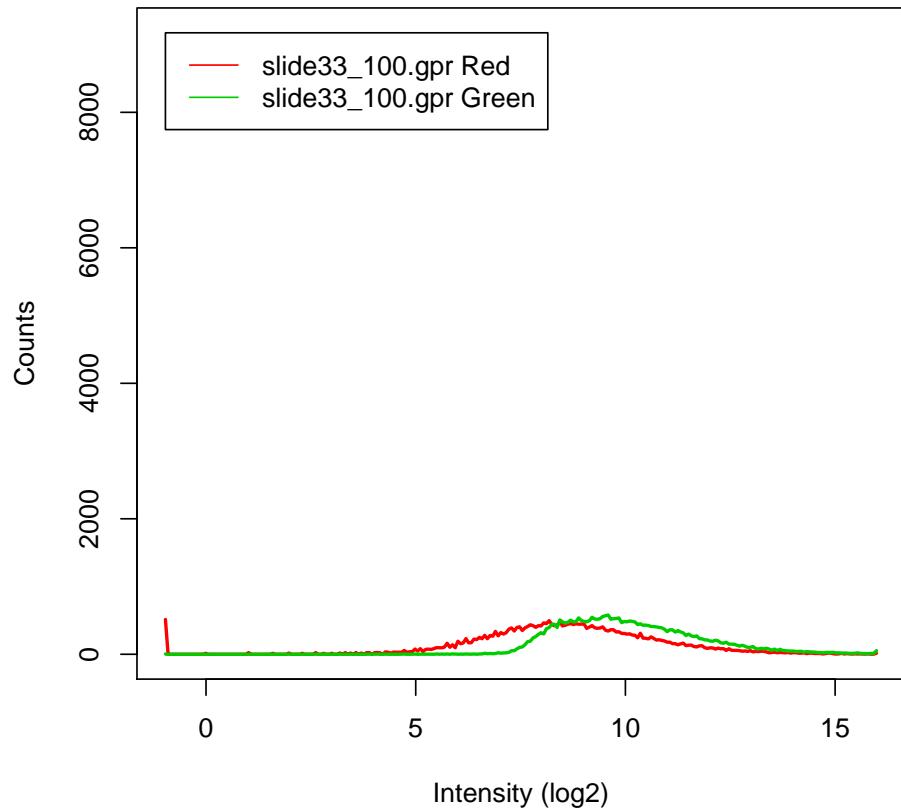


Figure 1.83: Histogram of the array 17 (slide33_100.gpr). Raw data after background correction.

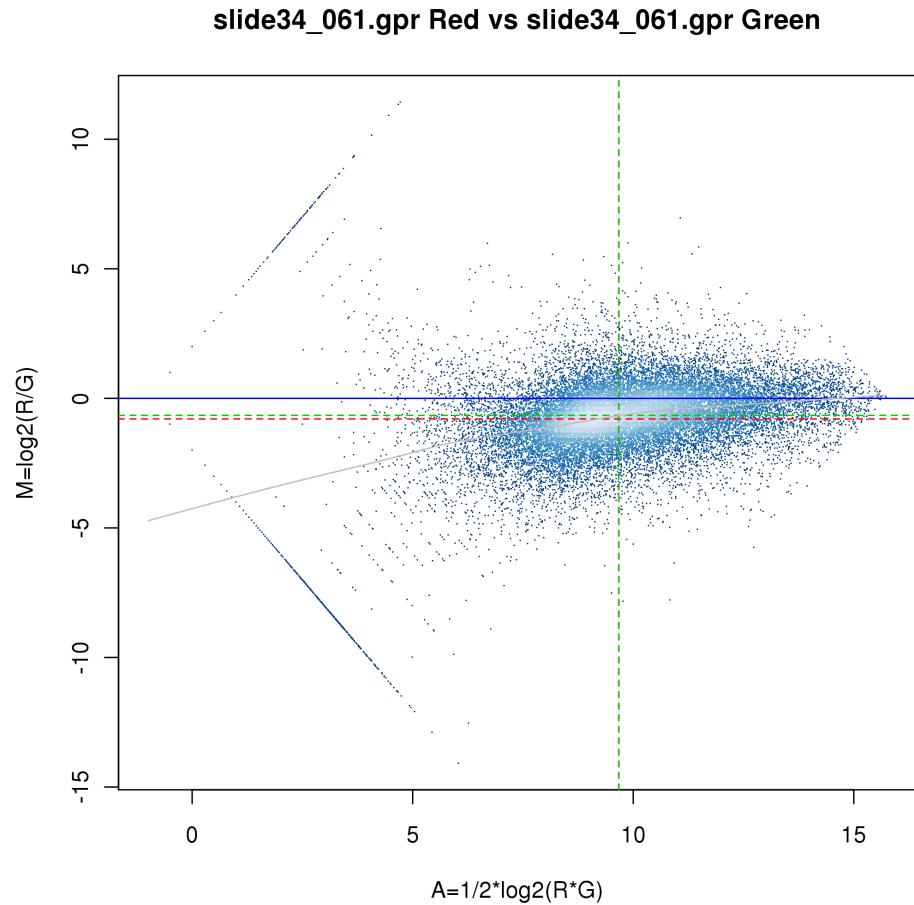


Figure 1.84: MA plot of array 18 (slide34_061.gpr). Raw data after background correction.

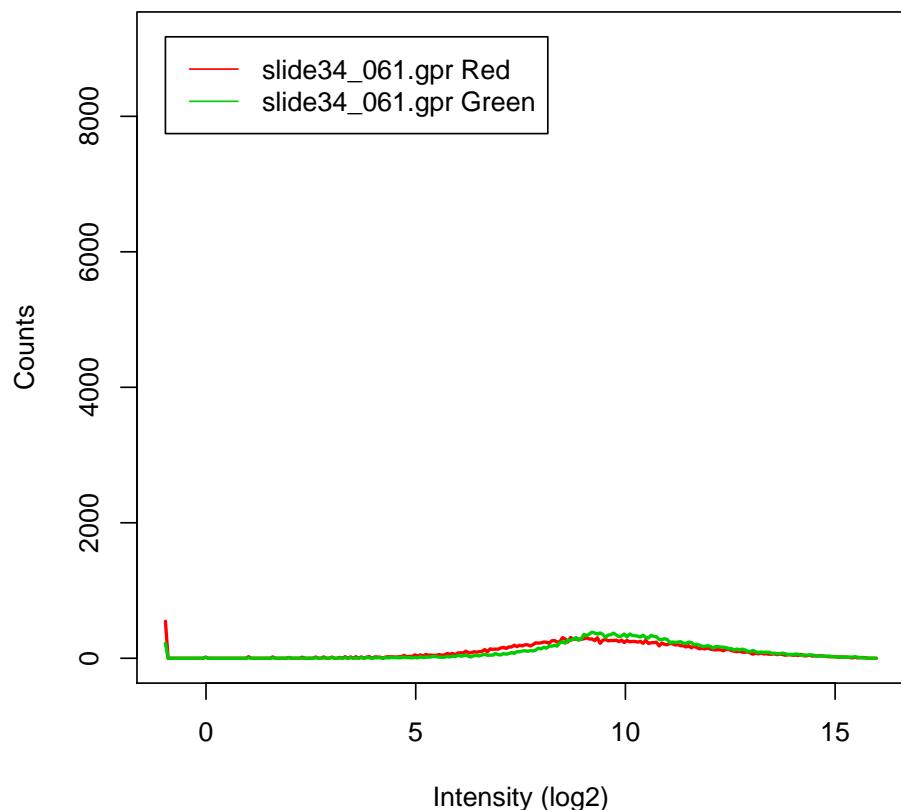


Figure 1.85: Histogram of the array 18 (slide34_061.gpr). Raw data after background correction.

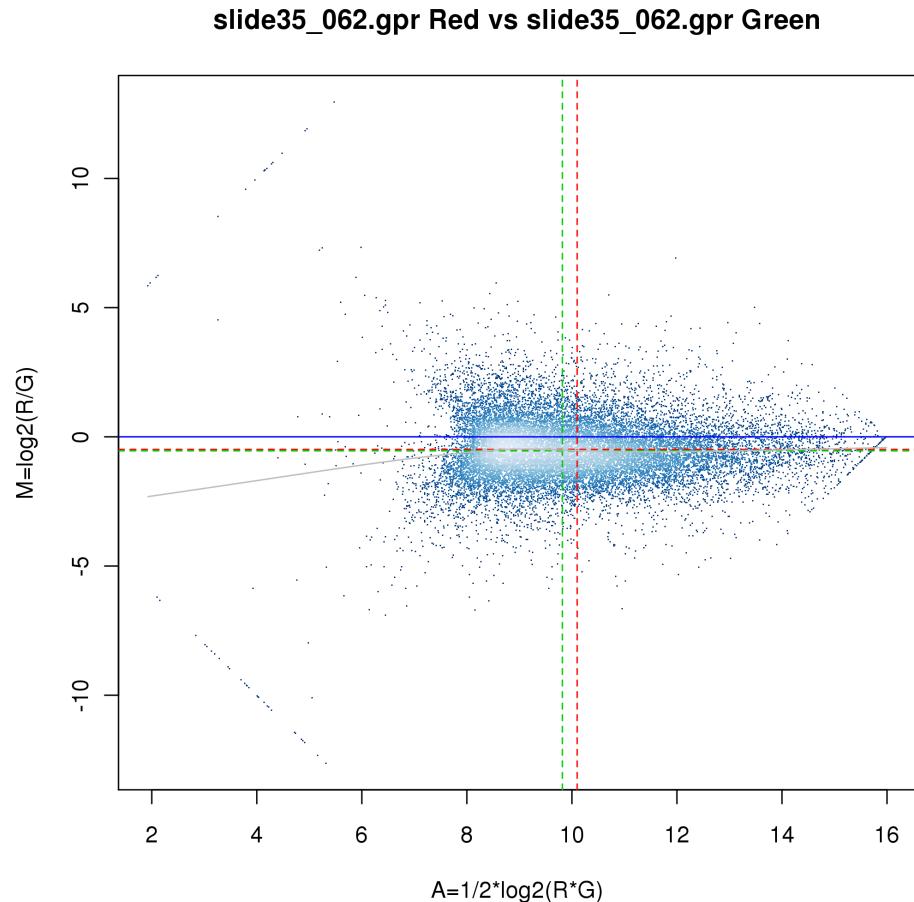


Figure 1.86: MA plot of array 19 (slide35_062.gpr). Raw data after background correction.

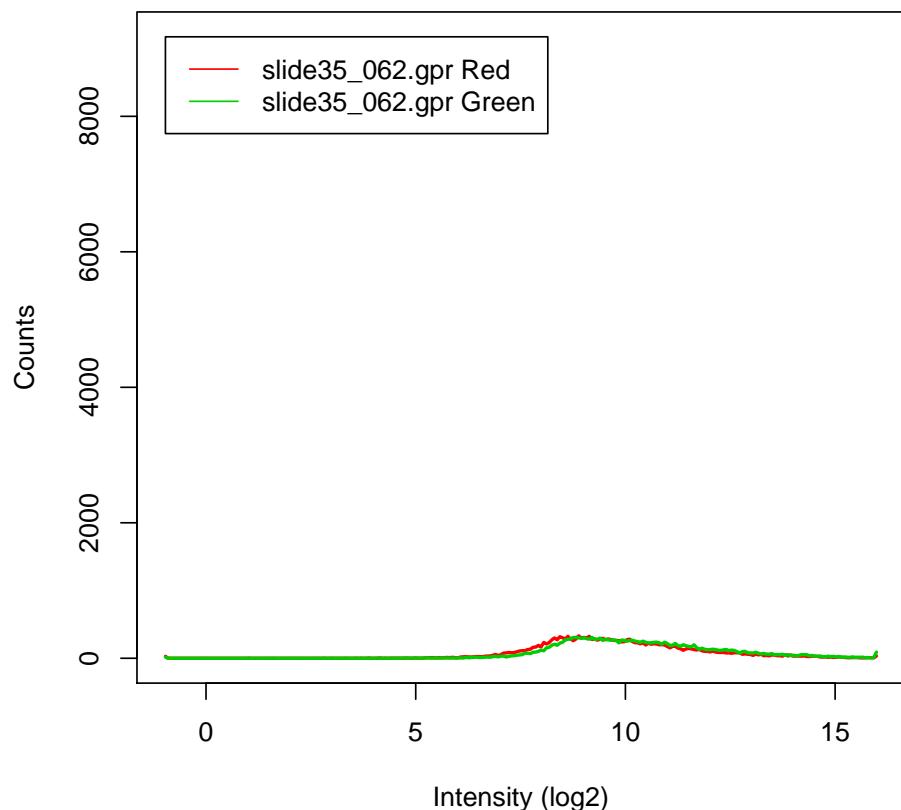


Figure 1.87: Histogram of the array 19 (slide35_062.gpr). Raw data after background correction.

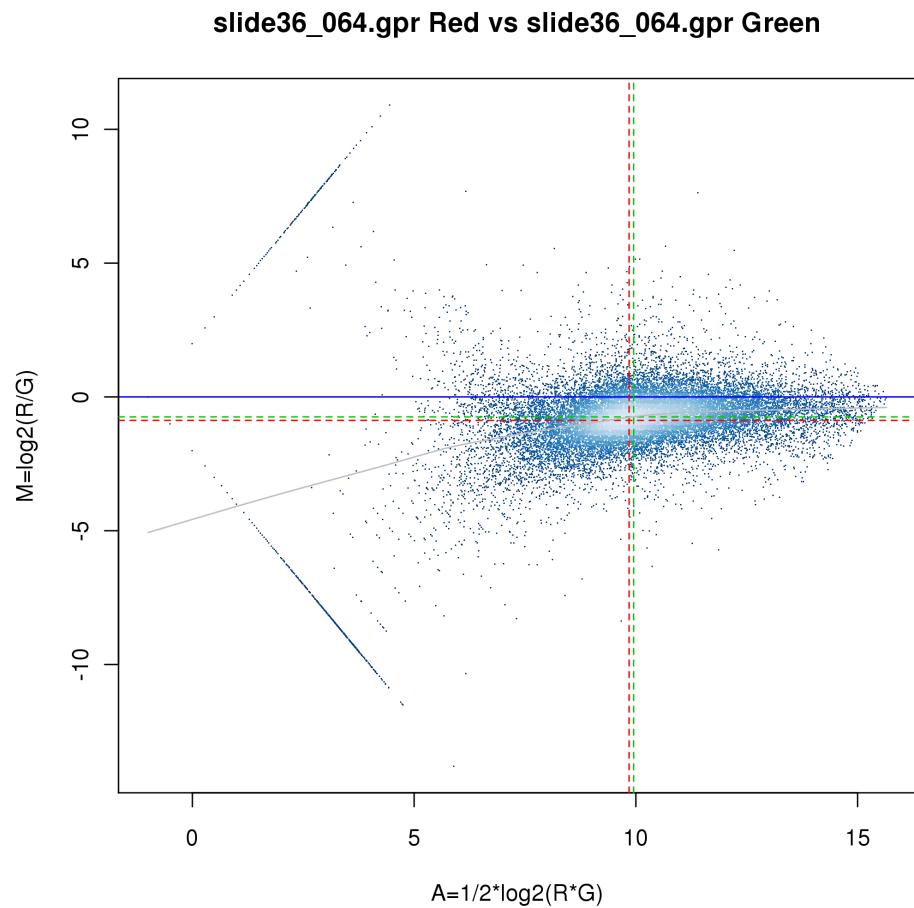


Figure 1.88: MA plot of array 20 (slide36_064.gpr). Raw data after background correction.

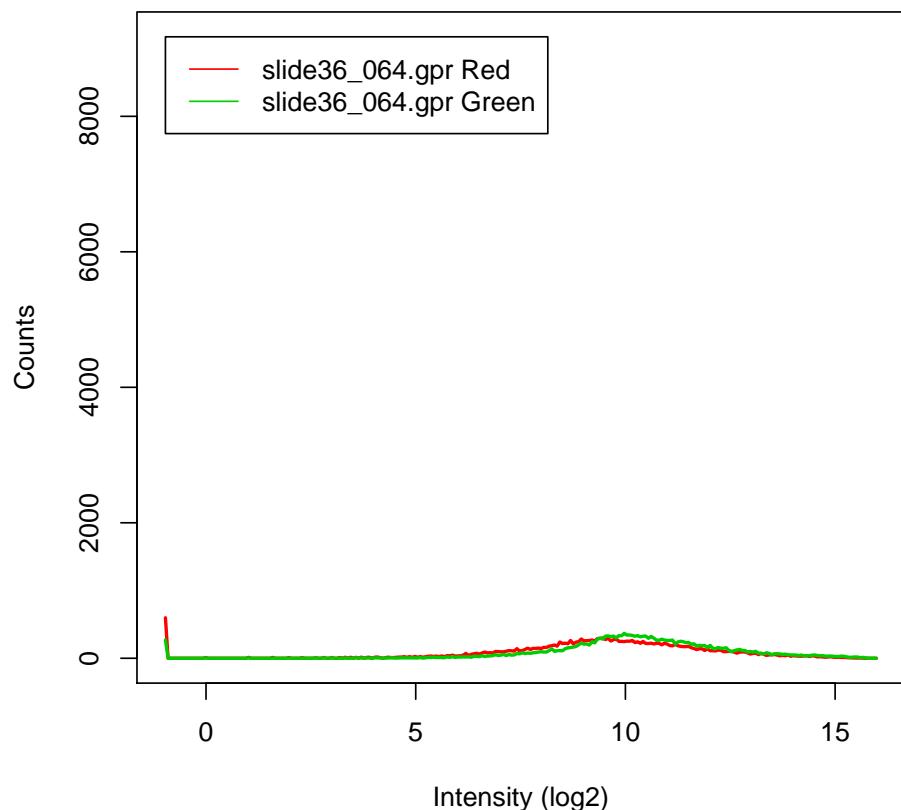


Figure 1.89: Histogram of the array 20 (slide36_064.gpr). Raw data after background correction.

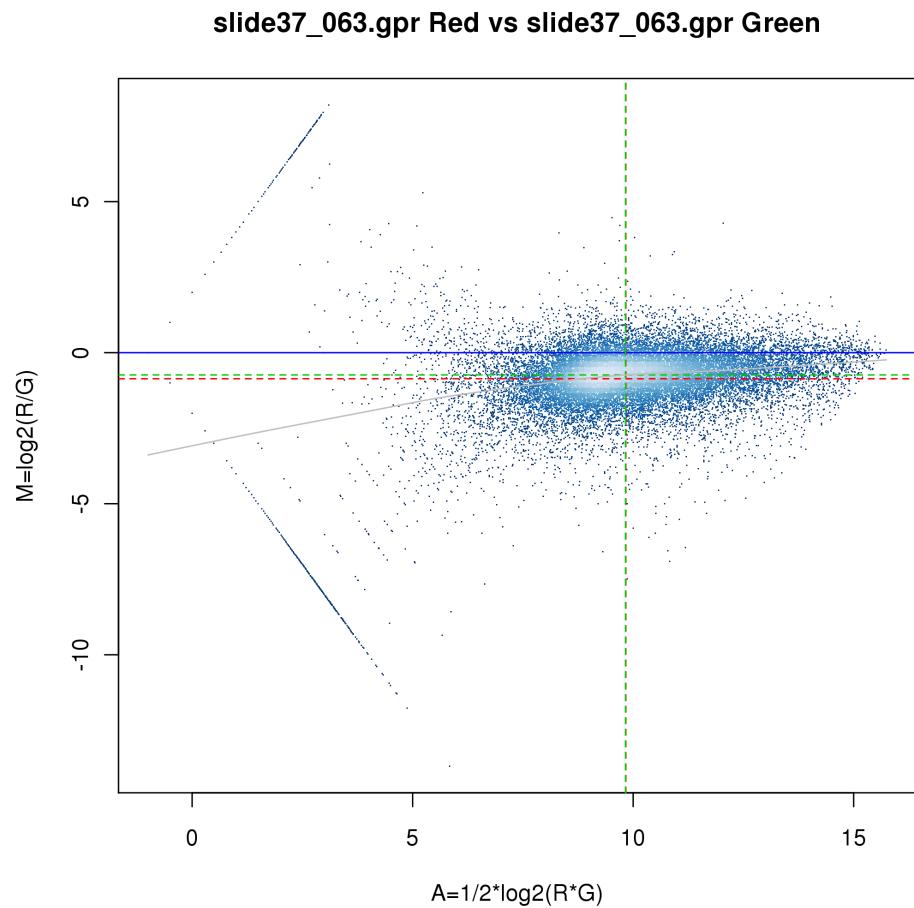


Figure 1.90: MA plot of array 21 (slide37_063.gpr). Raw data after background correction.

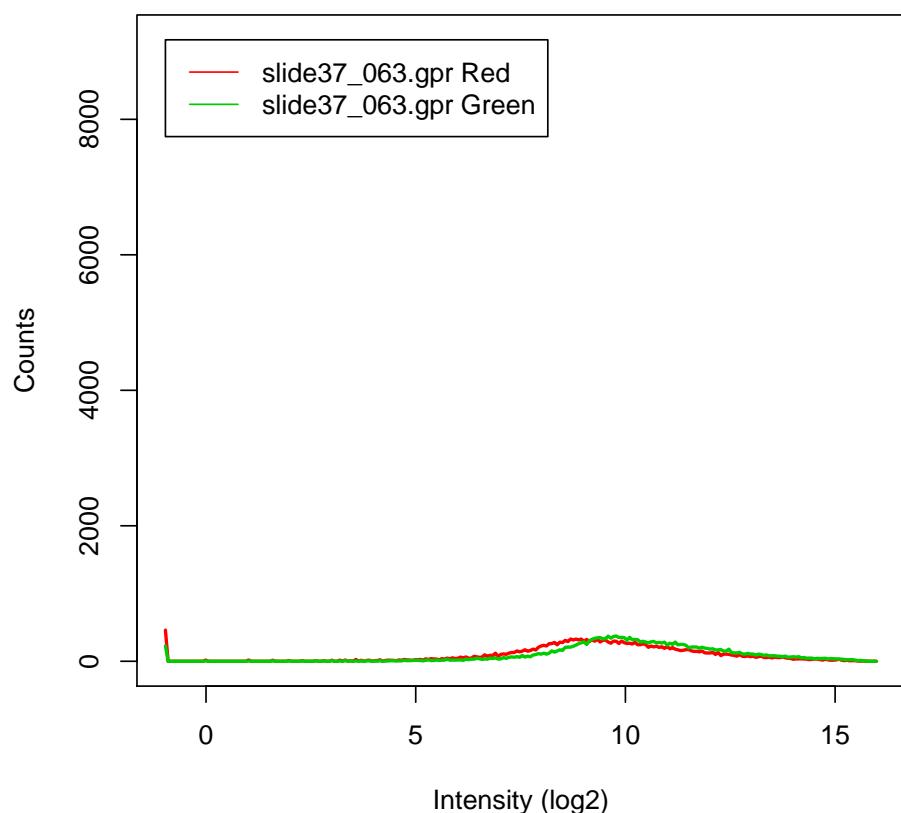


Figure 1.91: Histogram of the array 21 (slide37_063.gpr). Raw data after background correction.

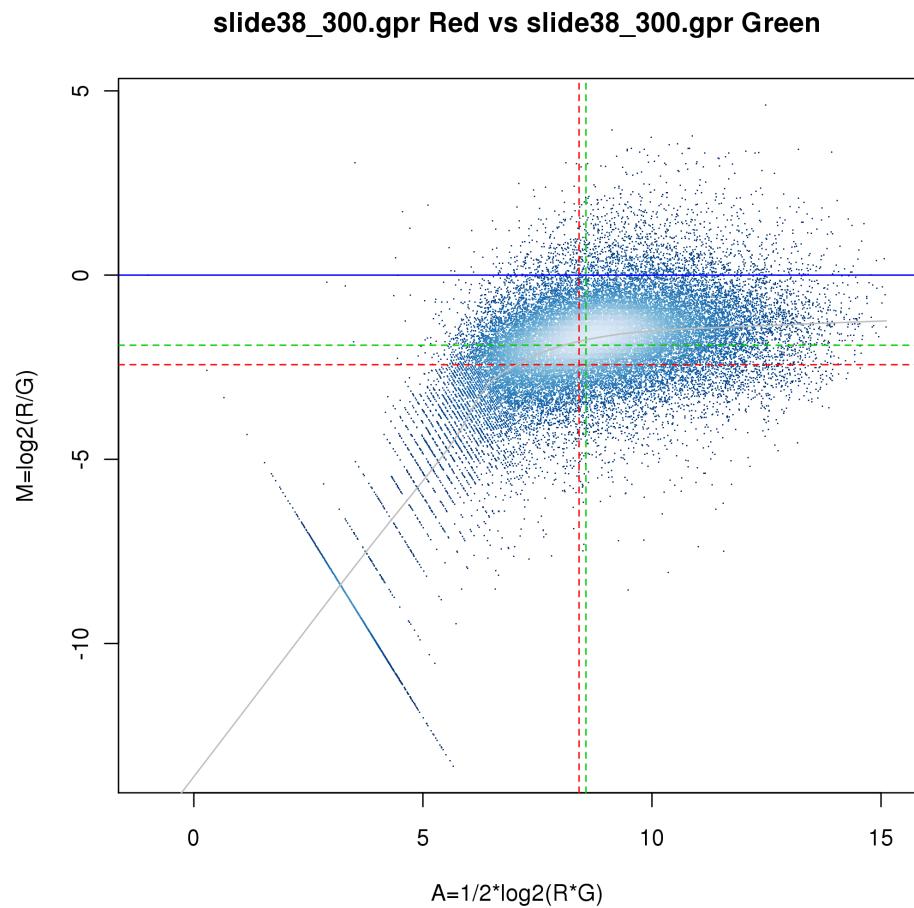


Figure 1.92: MA plot of array 22 (slide38_300.gpr). Raw data after background correction.

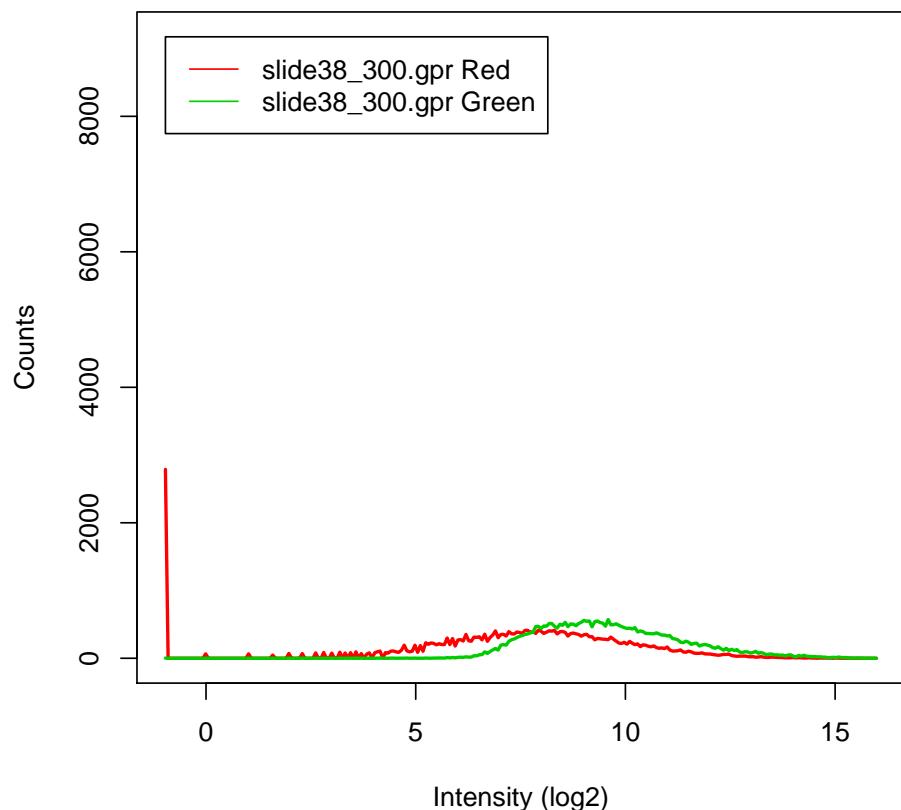


Figure 1.93: Histogram of the array 22 (slide38_300.gpr). Raw data after background correction.

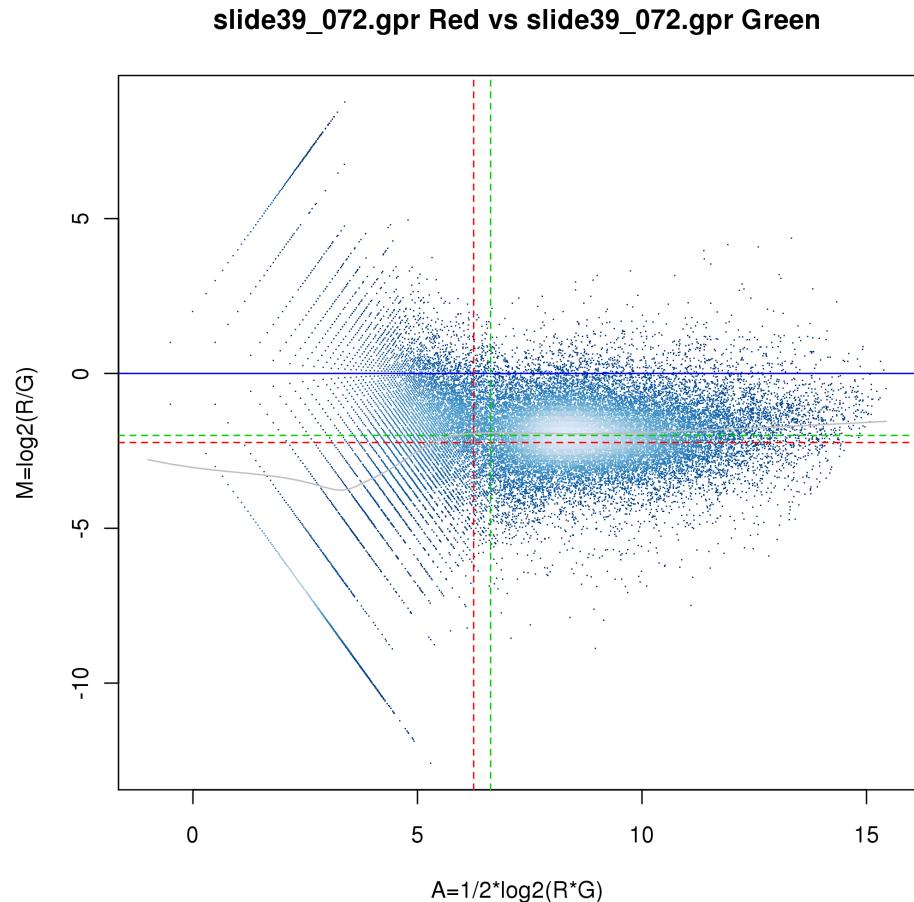


Figure 1.94: MA plot of array 23 (slide39_072.gpr). Raw data after background correction.

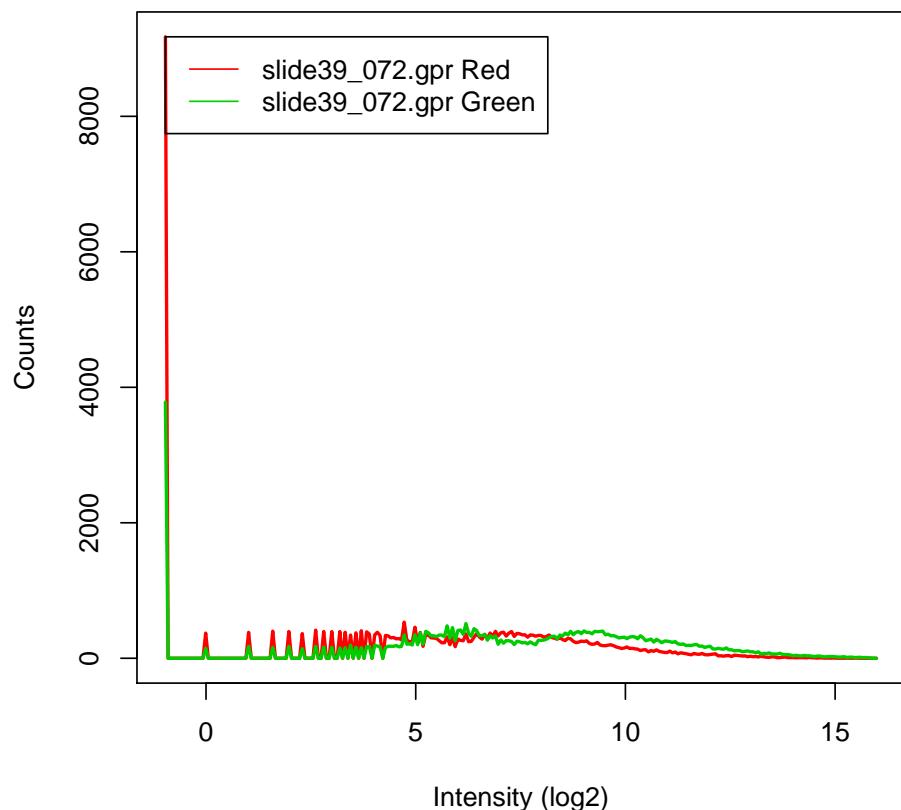


Figure 1.95: Histogram of the array 23 (slide39_072.gpr). Raw data after background correction.

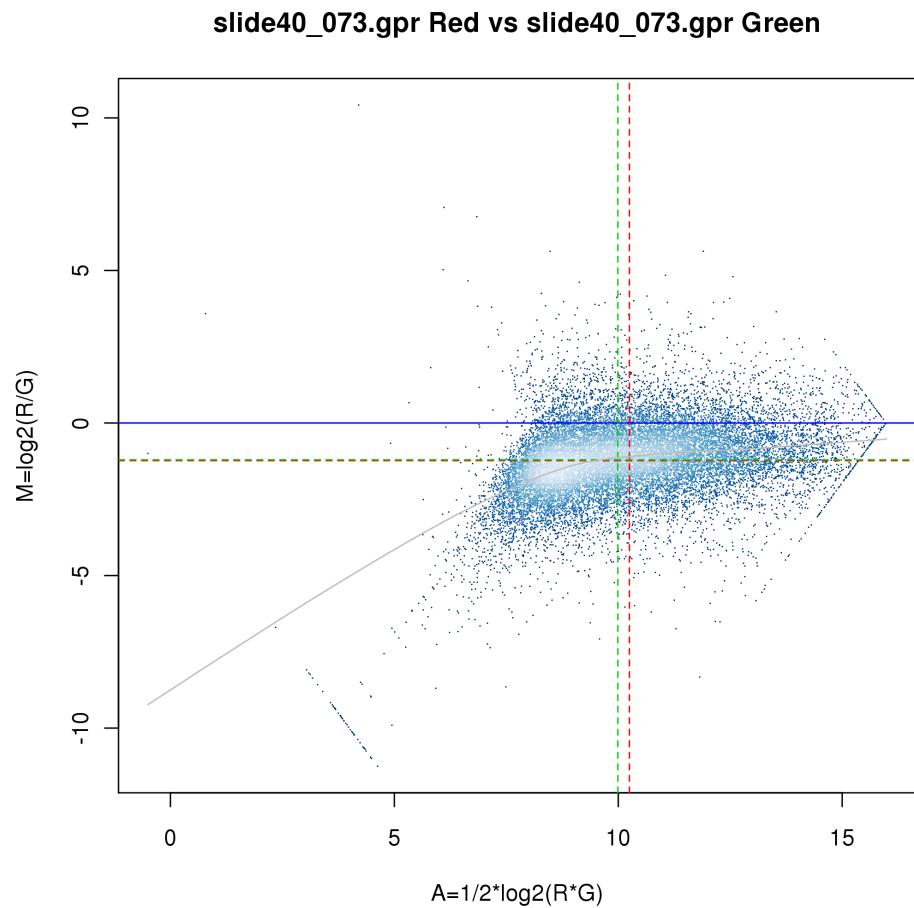


Figure 1.96: MA plot of array 24 (slide40_073.gpr). Raw data after background correction.

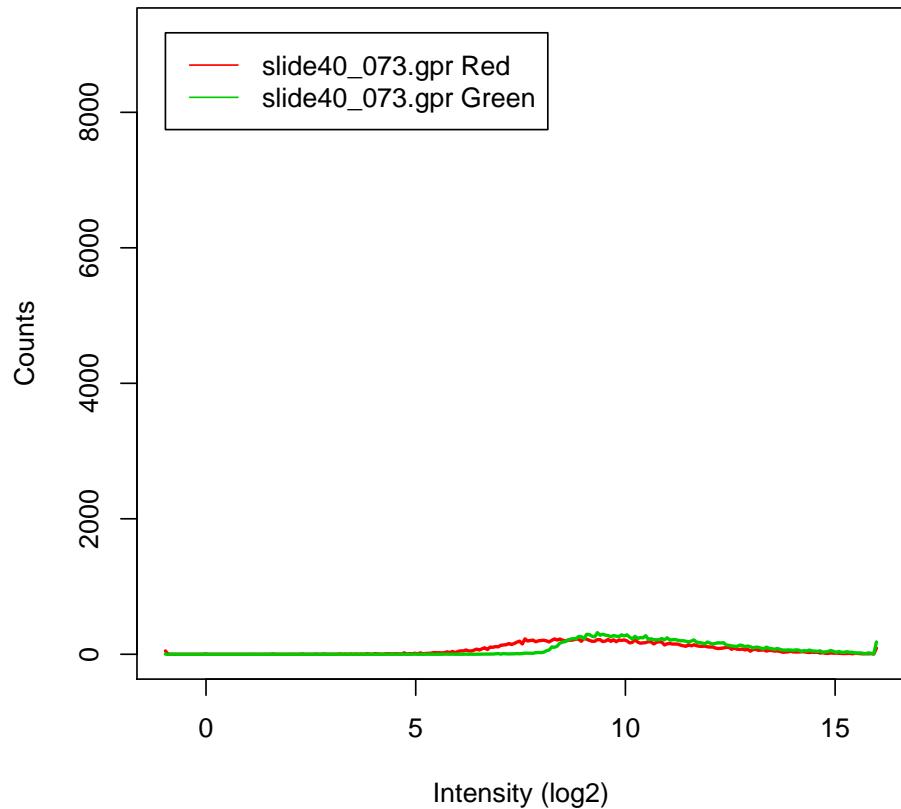


Figure 1.97: Histogram of the array 24 (slide40_073.gpr). Raw data after background correction.

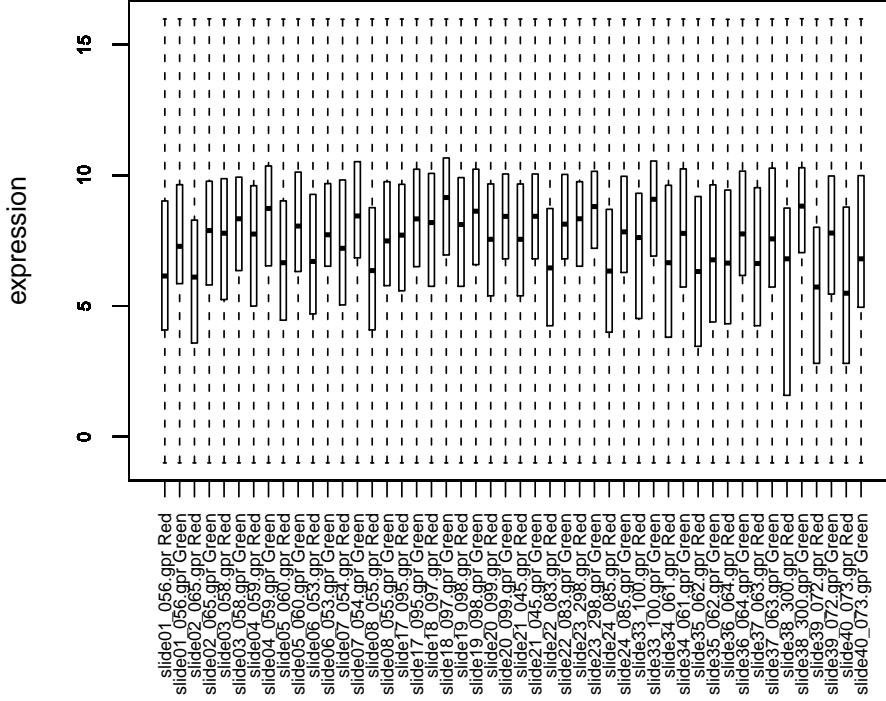


Figure 1.98: Boxplots of the signal intensities of each signal channel of the microarrays. Raw data after background correction.

1.3 Within array normalization

Normalization is intended to remove from the expression measures any systematic trends which arise from the microarray technology rather than from differences between the probes or between the target RNA samples hybridized to the arrays.

```
> Method <- "printtiploess"
> if (is.null(Slides.raw$printer)) {
+   Method <- "loess"
+ }
> Slides.norm <- normalizeWithinArrays(Slides.raw, layout = Slides.raw$printer,
+   method = Method)
> rm(Slides.raw)
> g <- gc()
```

Next diagnostic plots of the (within array) normalized data will be drawn.

```
> Dummy <- newMadbSet(Slides.norm)

Converting a limma MAList into a MadbSet...
Setting the weights... a weights of 0 means the gene was flagged, a weights of one means the signal is ok!

Inserting available annotation into the slot @genes

Inserting available annotation into the slot @genes
```

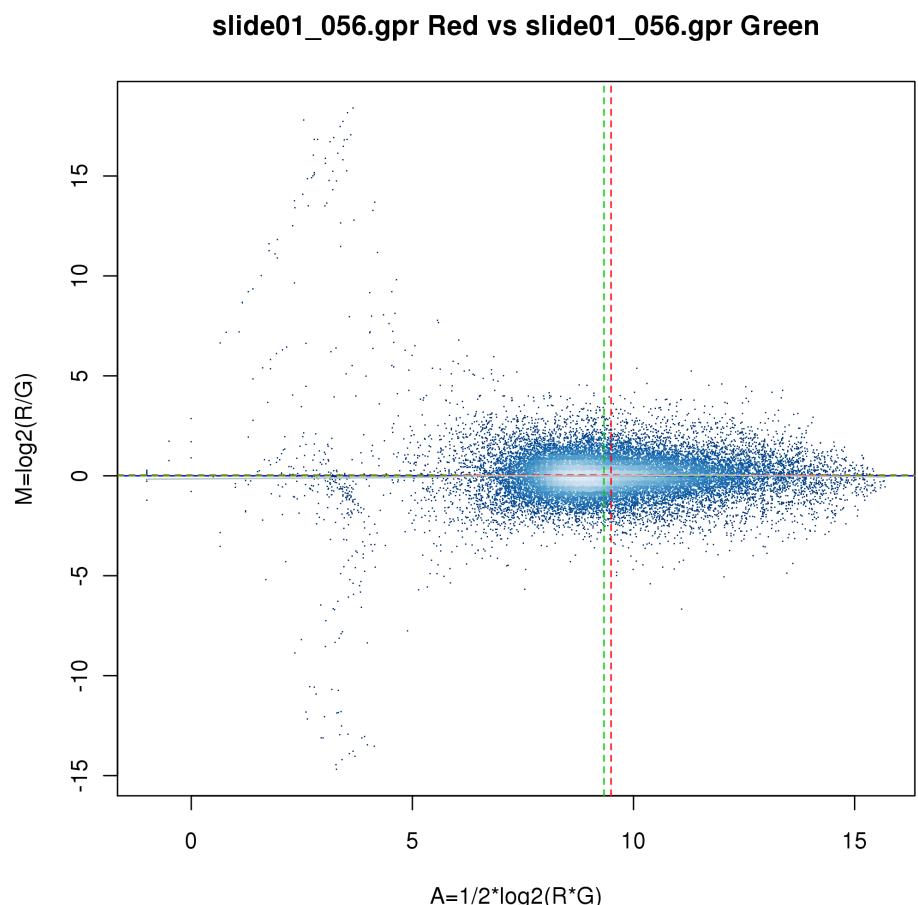


Figure 1.99: MA plot of array 1 (slide01_056.gpr). Within array normalized data.

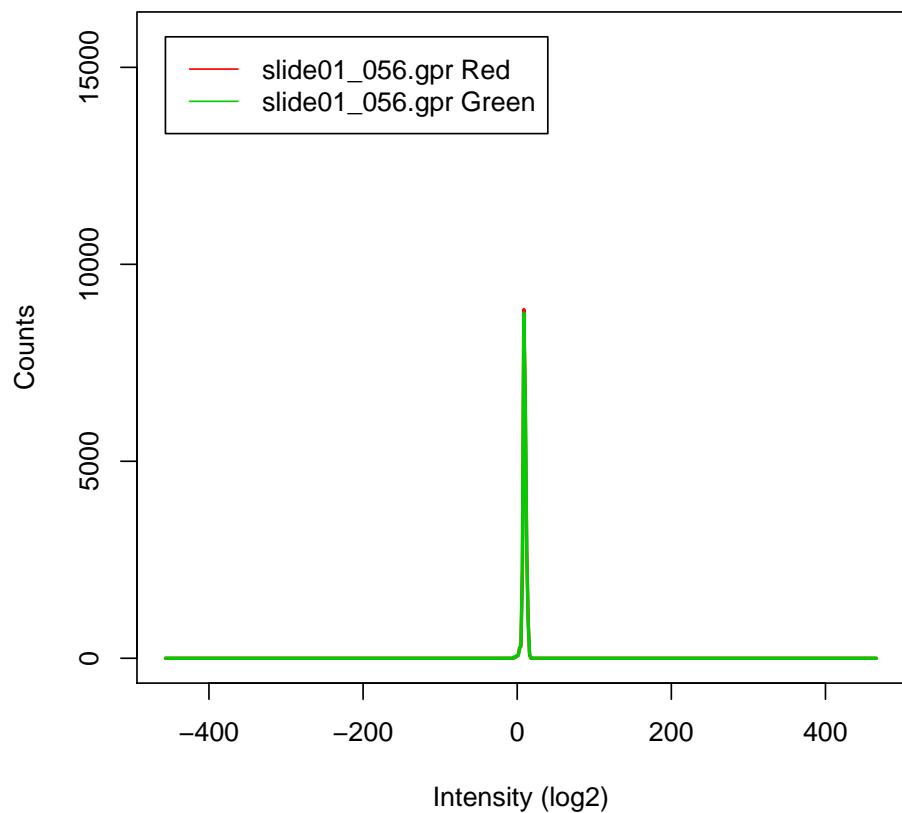


Figure 1.100: Histogram of the array 1 (slide01_056.gpr). Within array normalized data.

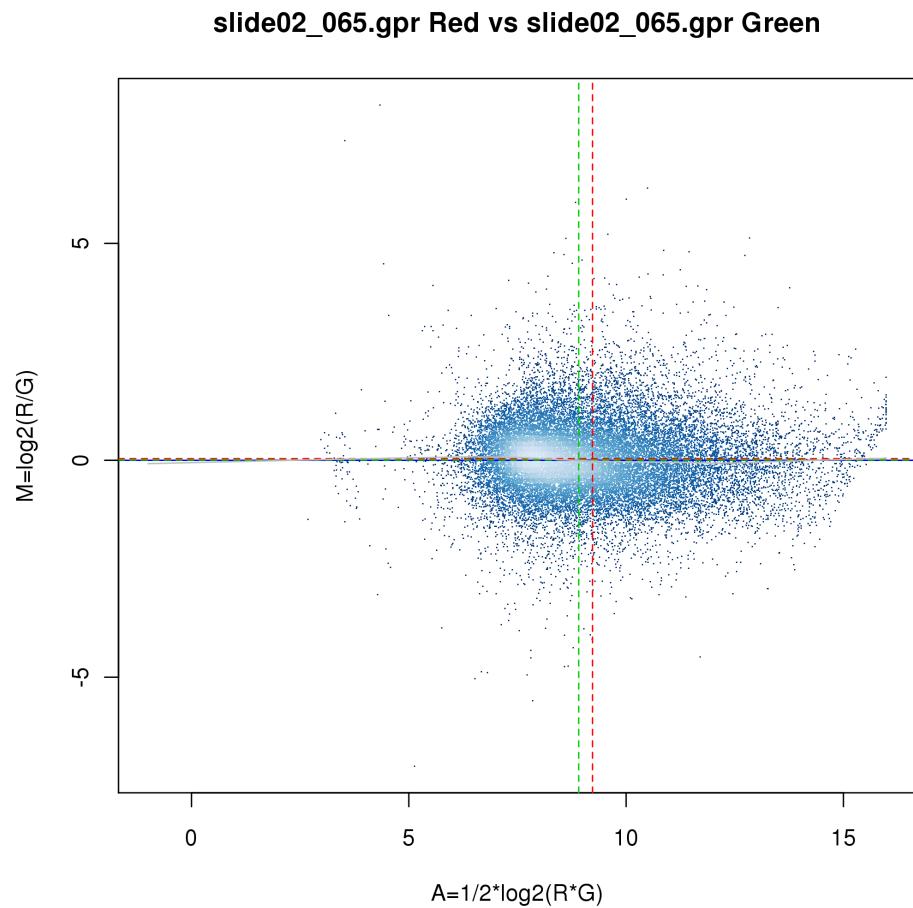


Figure 1.101: MA plot of array 2 (slide02_065.gpr). Within array normalized data.

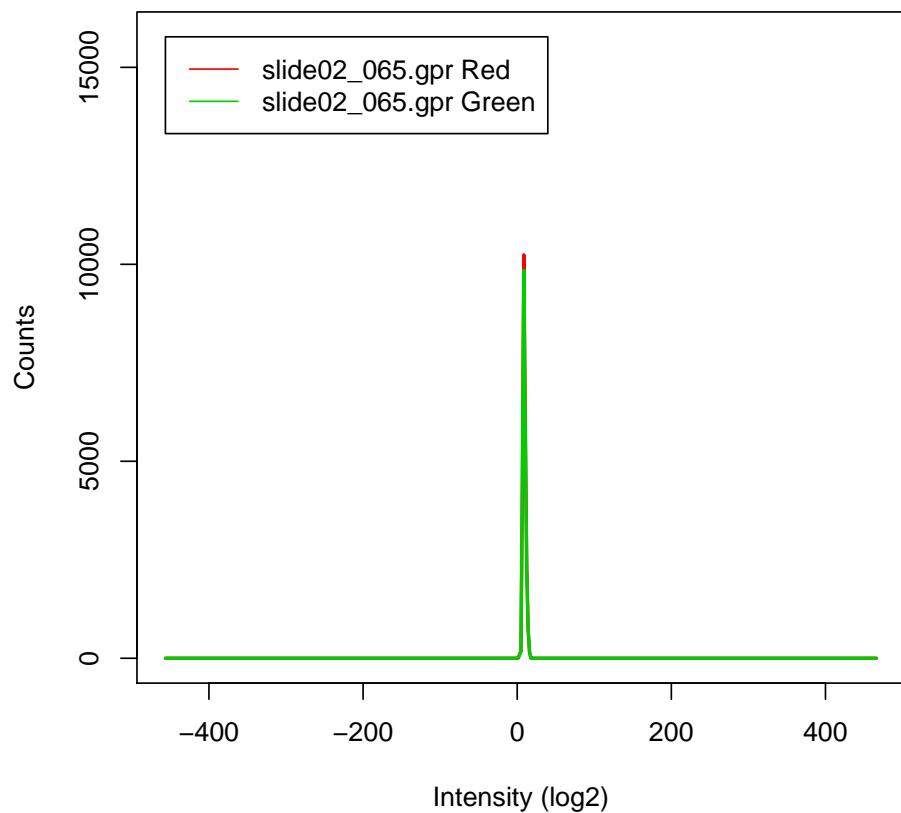


Figure 1.102: Histogram of the array 2 (slide02_065.gpr). Within array normalized data.

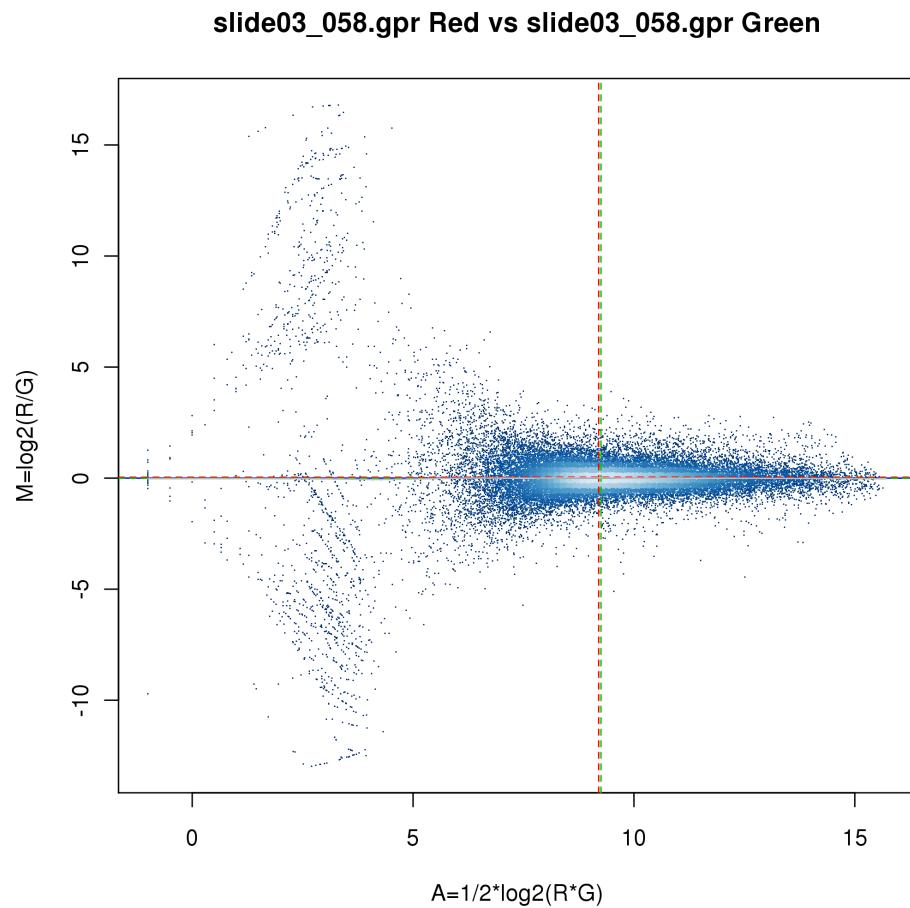


Figure 1.103: MA plot of array 3 (slide03_058.gpr). Within array normalized data.

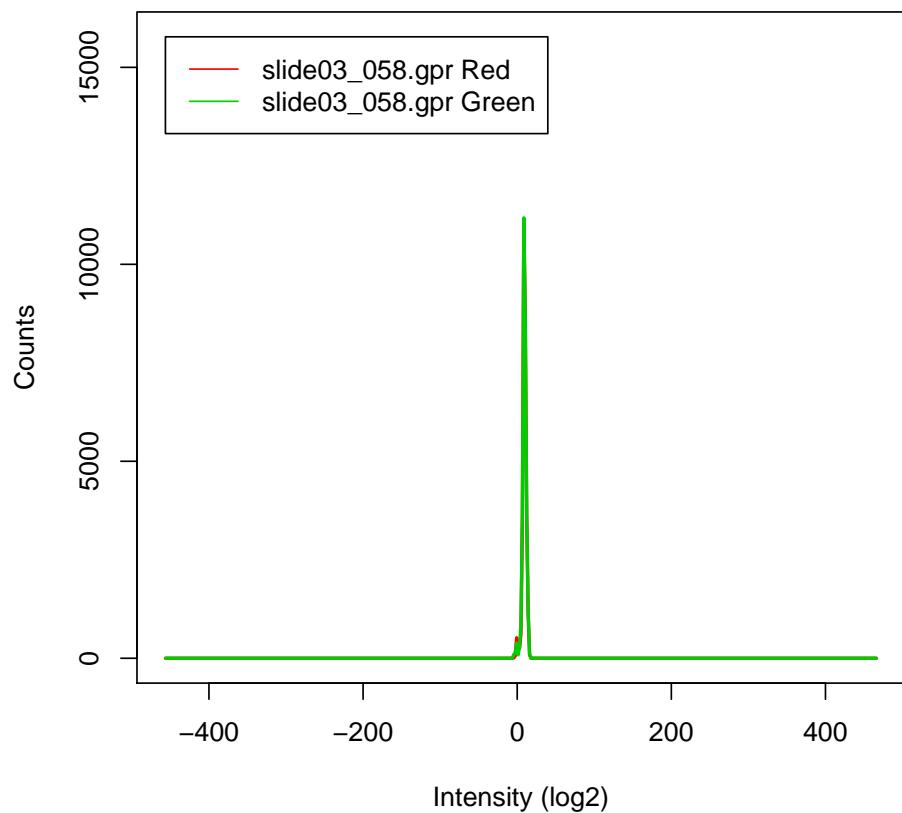


Figure 1.104: Histogram of the array 3 (slide03_058.gpr). Within array normalized data.

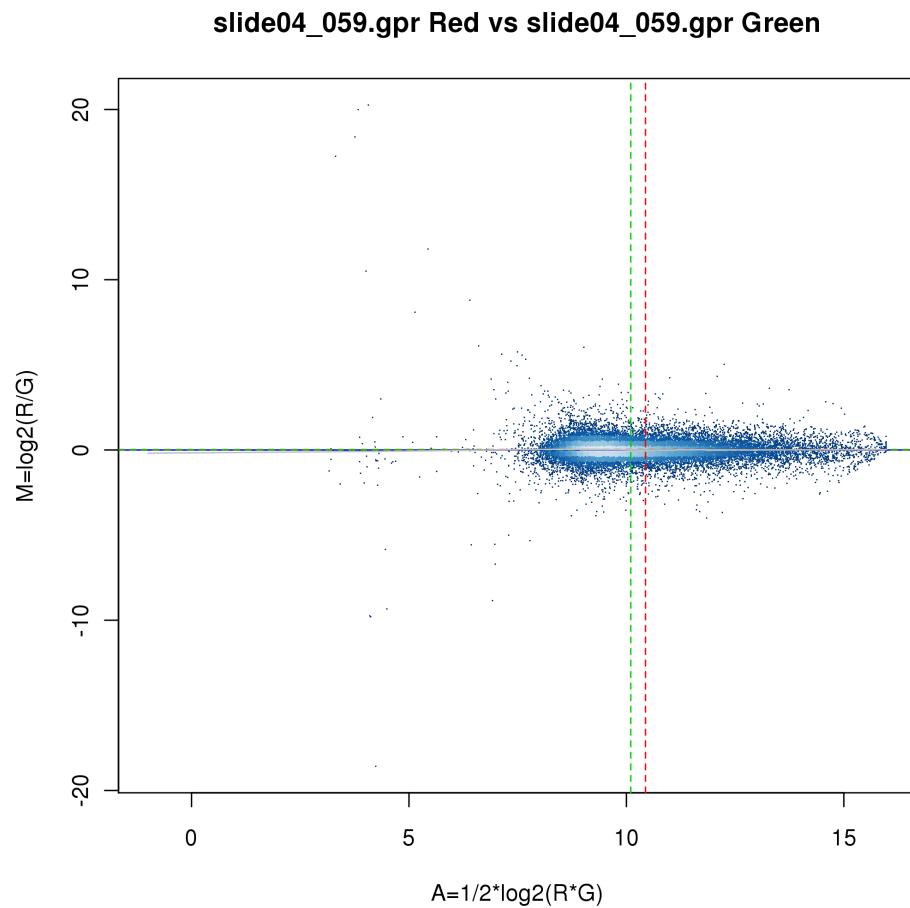


Figure 1.105: MA plot of array 4 (slide04_059.gpr). Within array normalized data.

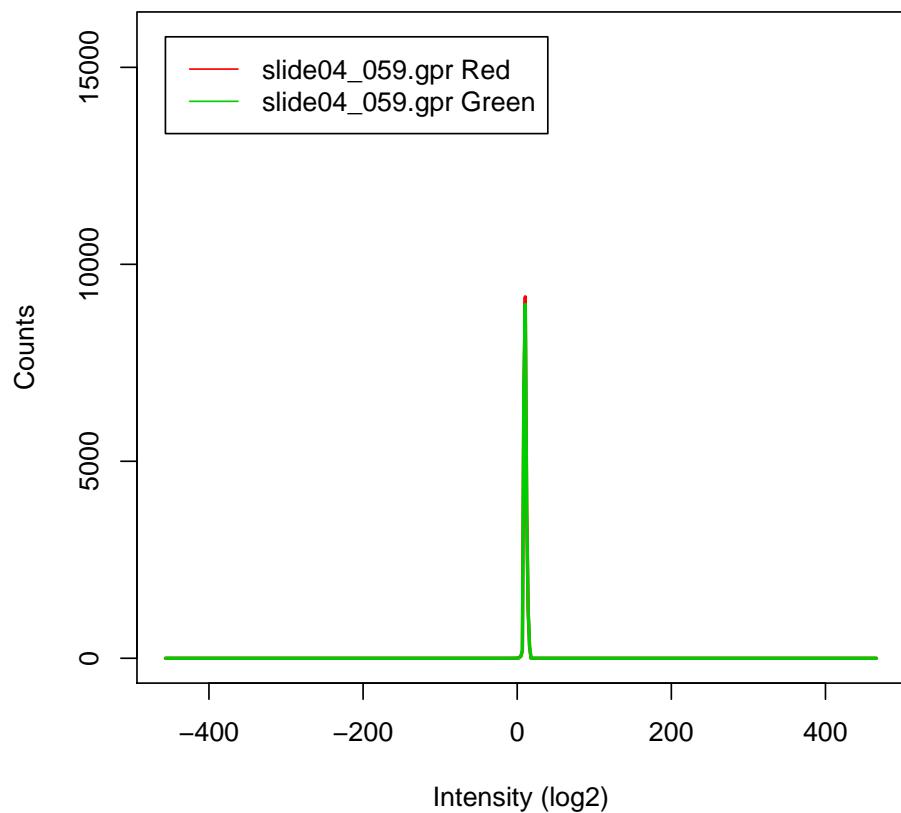


Figure 1.106: Histogram of the array 4 (slide04_059.gpr). Within array normalized data.

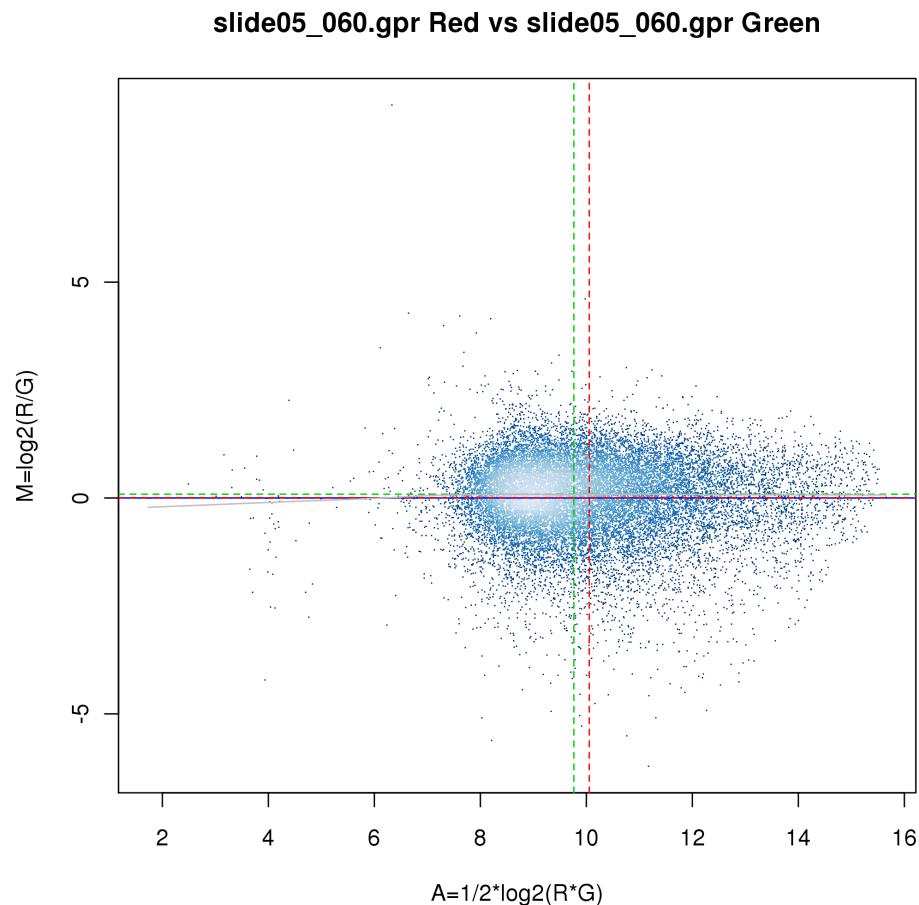


Figure 1.107: MA plot of array 5 (slide05_060.gpr). Within array normalized data.

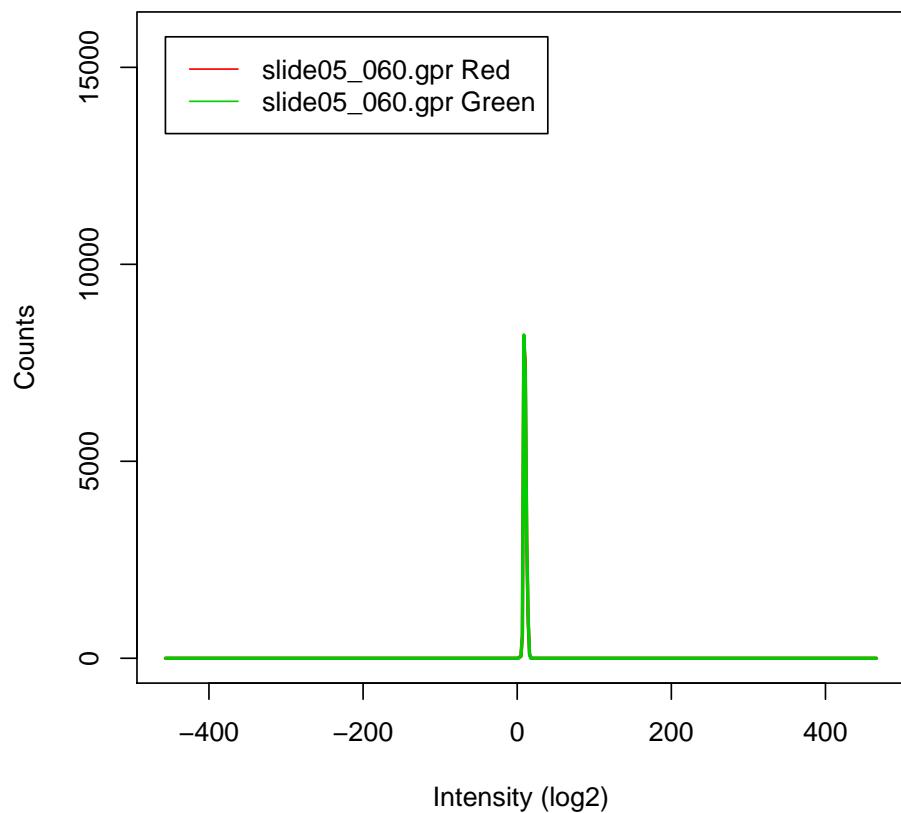


Figure 1.108: Histogram of the array 5 (slide05_060.gpr). Within array normalized data.

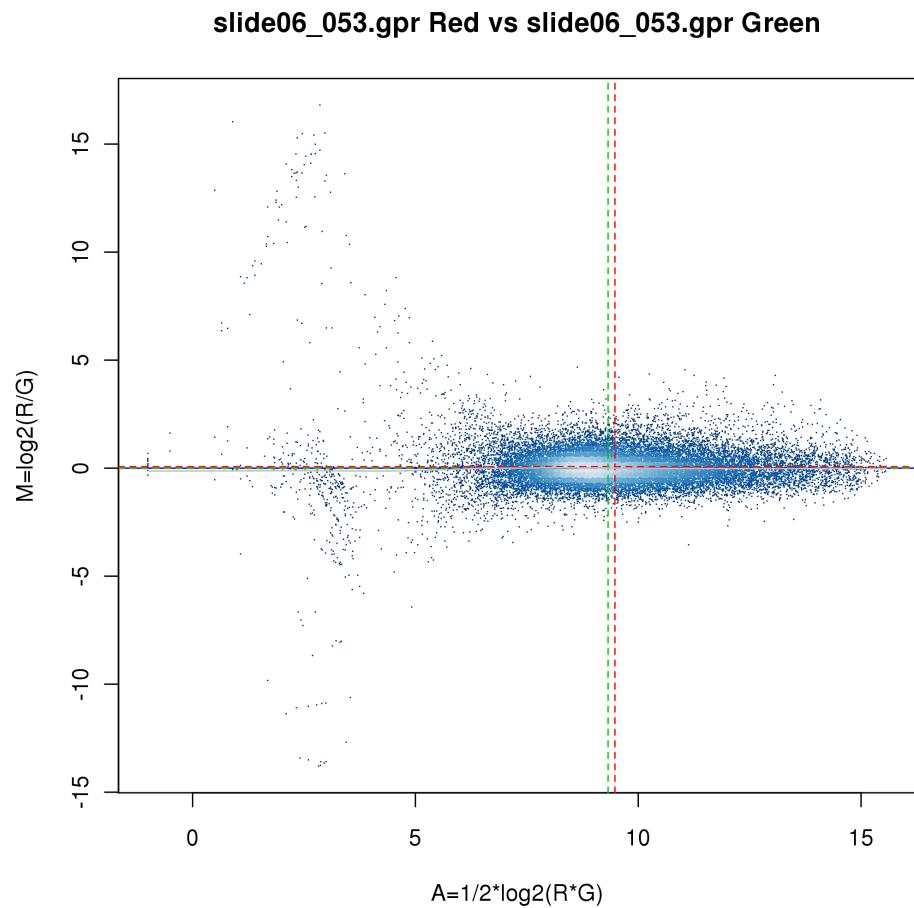


Figure 1.109: MA plot of array 6 (slide06_053.gpr). Within array normalized data.

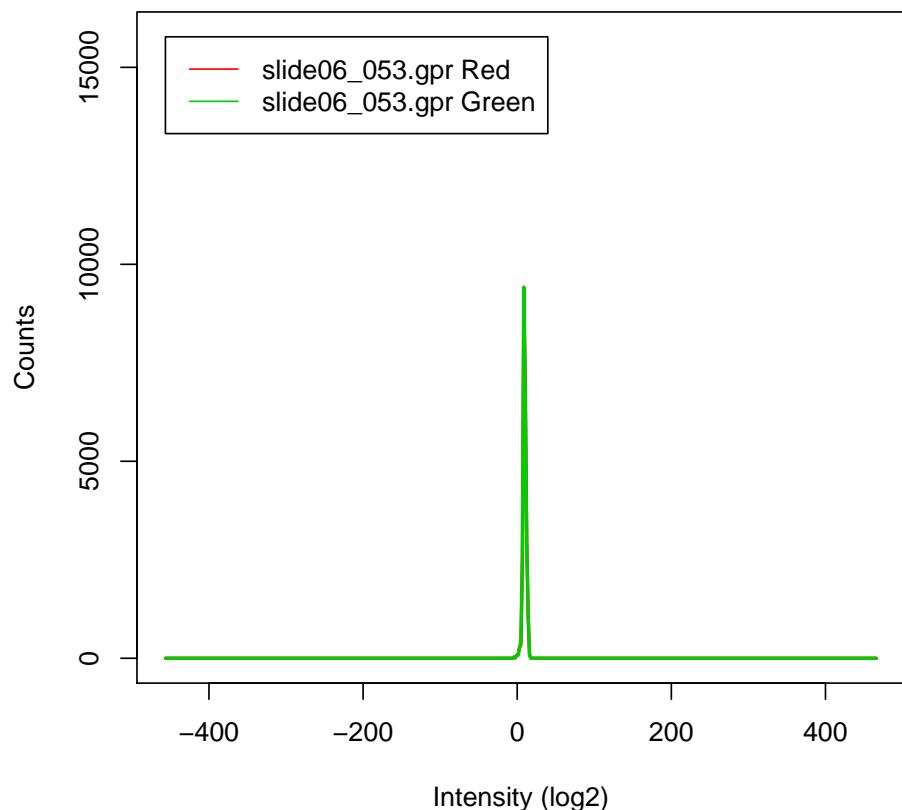


Figure 1.110: Histogram of the array 6 (slide06_053.gpr). Within array normalized data.

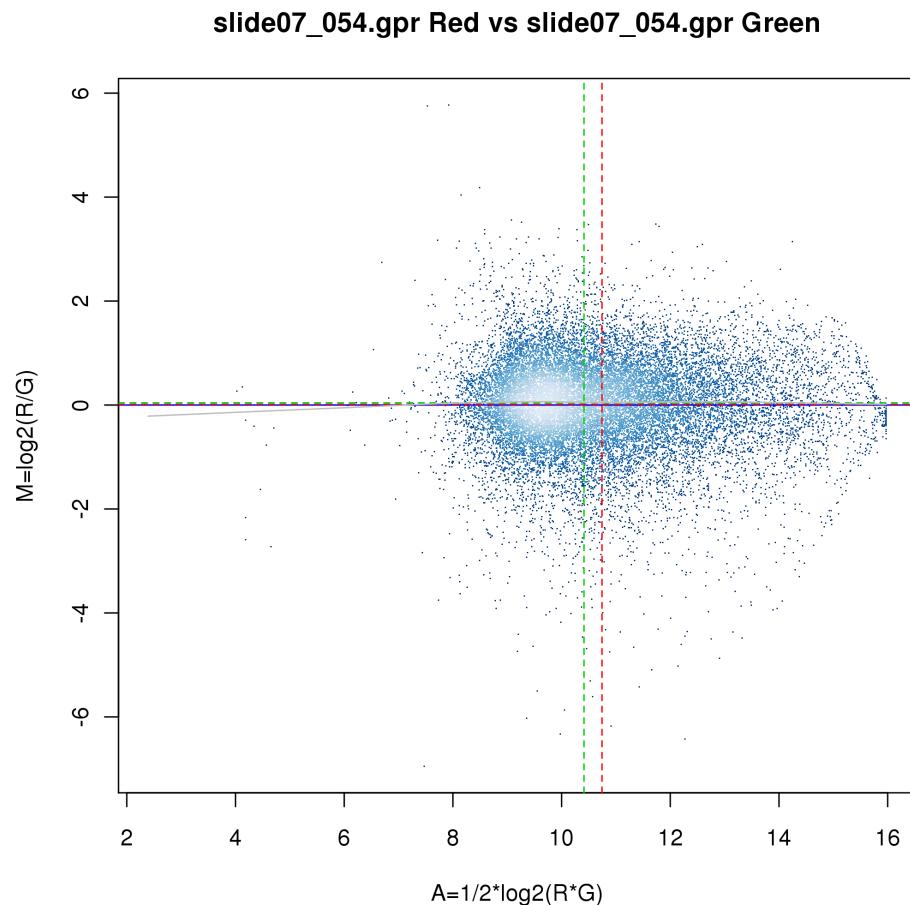


Figure 1.111: MA plot of array 7 (slide07_054.gpr). Within array normalized data.

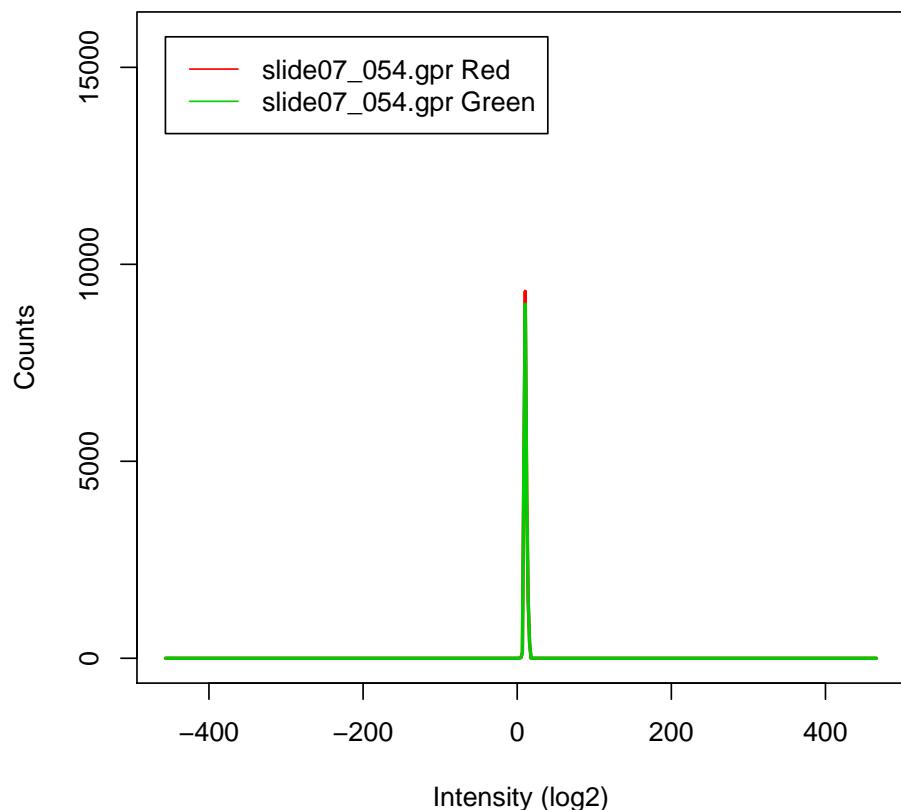


Figure 1.112: Histogram of the array 7 (slide07_054.gpr). Within array normalized data.

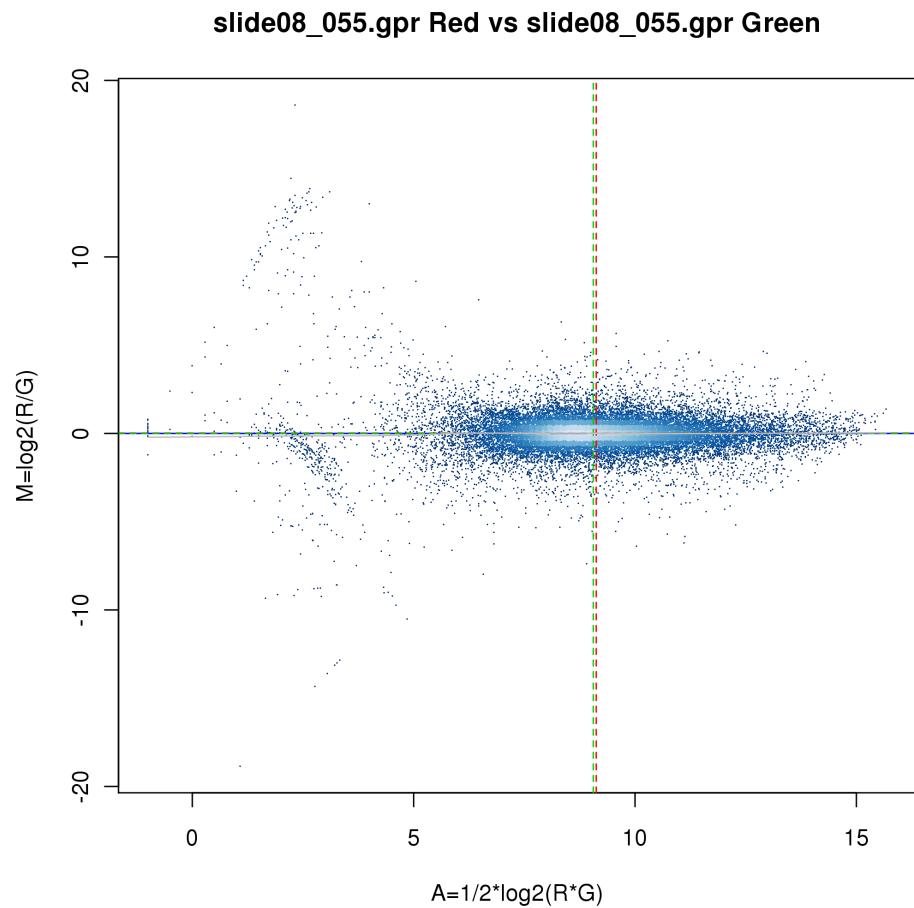


Figure 1.113: MA plot of array 8 (slide08_055.gpr). Within array normalized data.

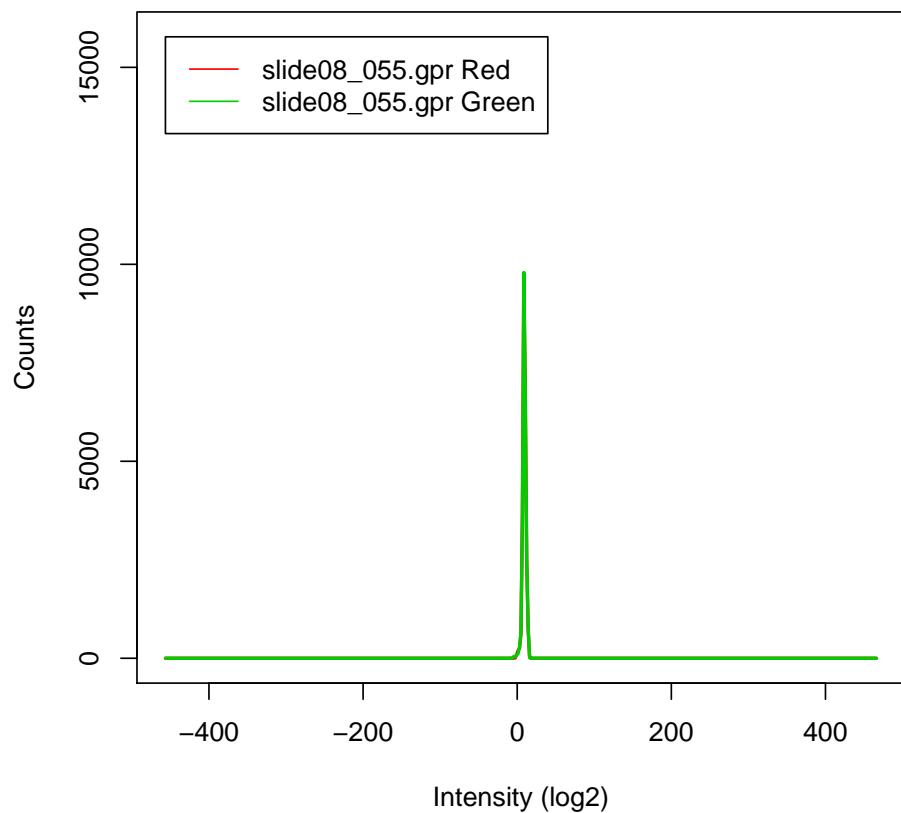


Figure 1.114: Histogram of the array 8 (slide08_055.gpr). Within array normalized data.

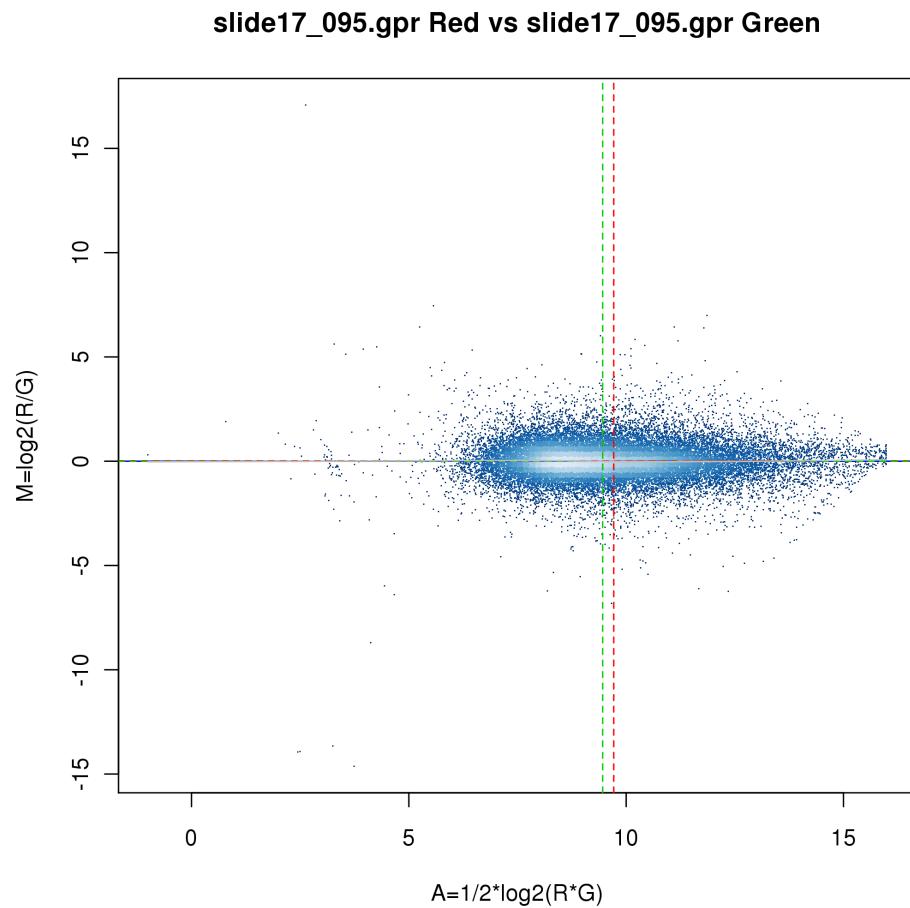


Figure 1.115: MA plot of array 9 (slide17_095.gpr). Within array normalized data.

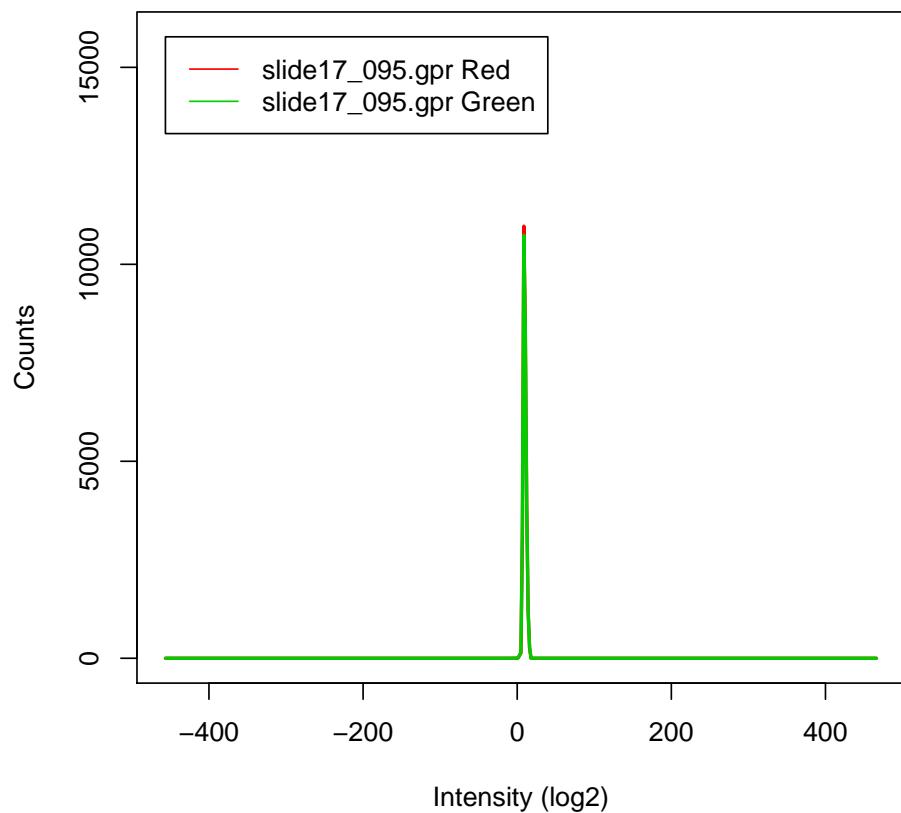


Figure 1.116: Histogram of the array 9 (slide17_095.gpr). Within array normalized data.

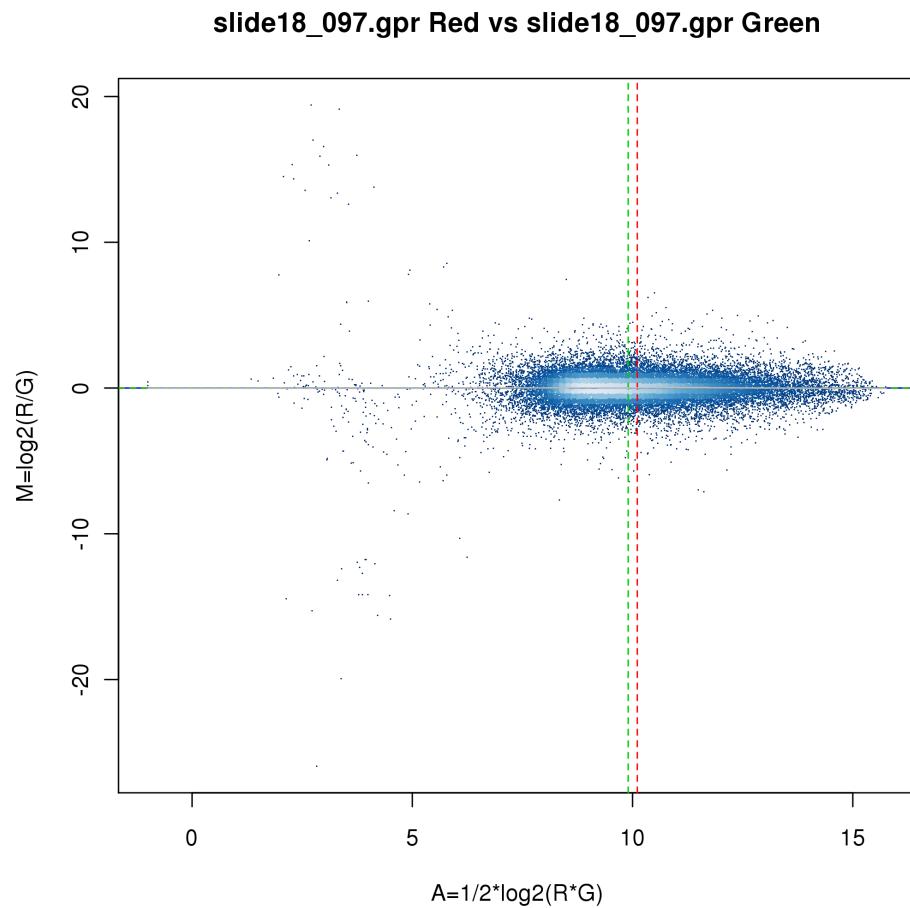


Figure 1.117: MA plot of array 10 (slide18_097.gpr). Within array normalized data.

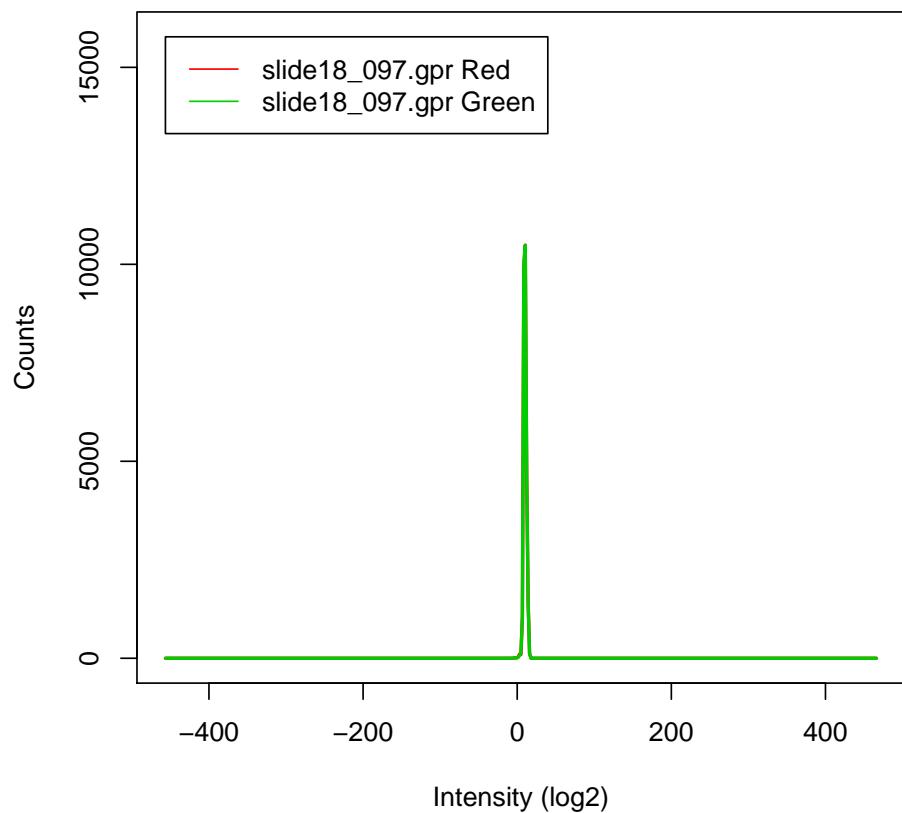


Figure 1.118: Histogram of the array 10 (slide18_097.gpr). Within array normalized data.

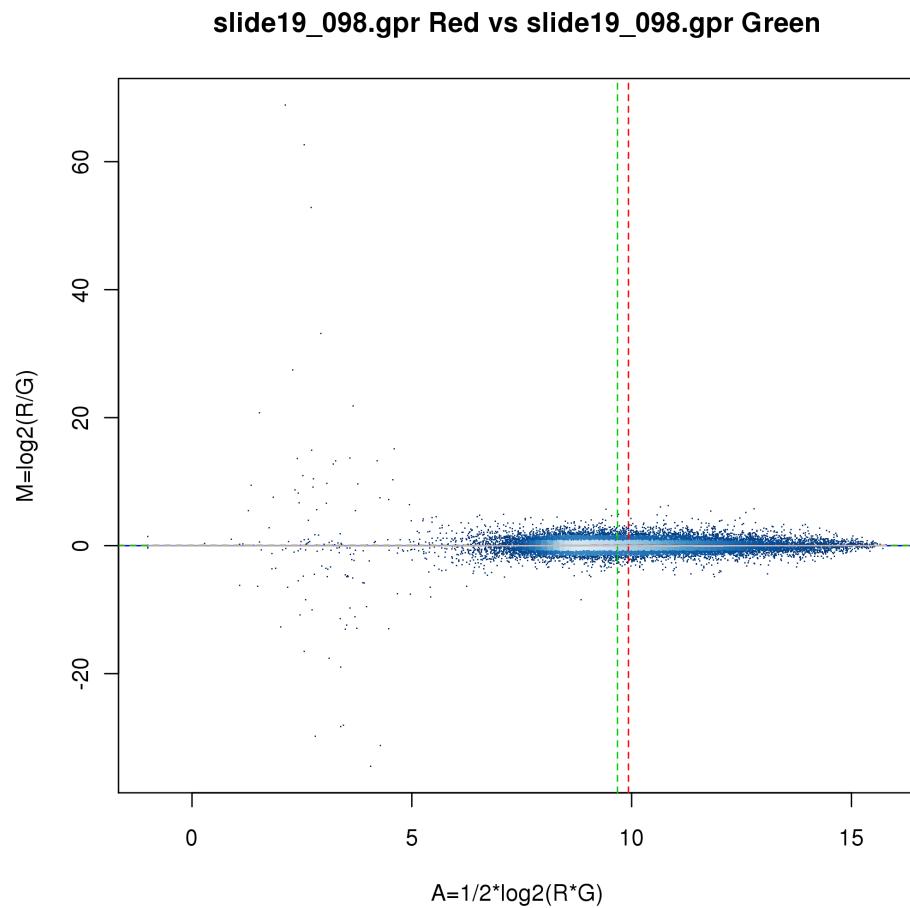


Figure 1.119: MA plot of array 11 (slide19_098.gpr). Within array normalized data.

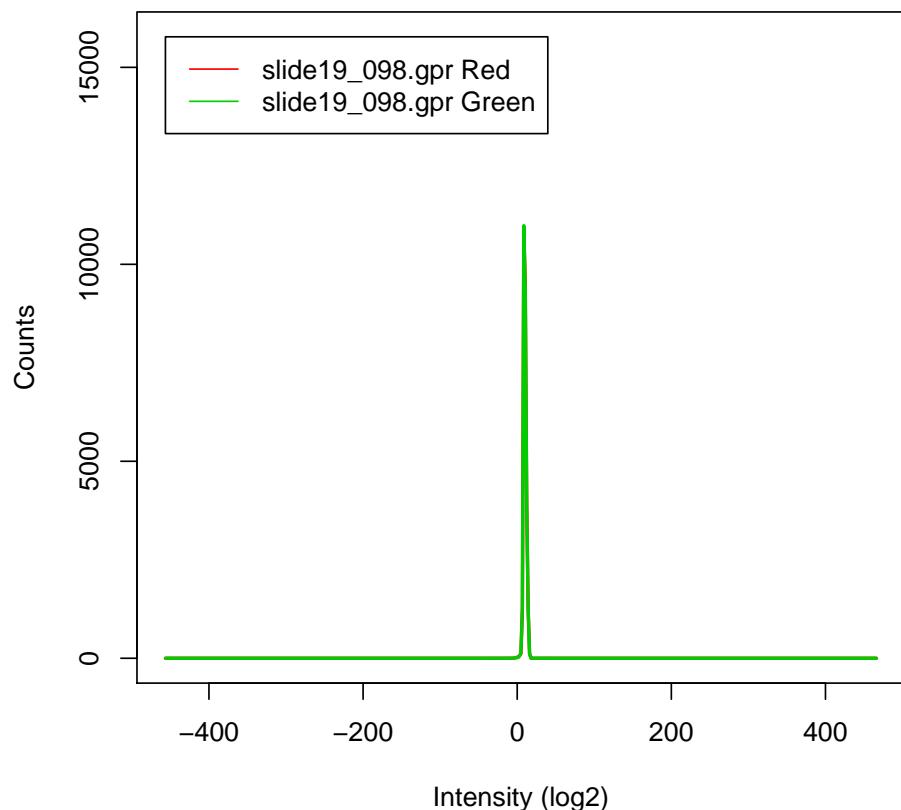


Figure 1.120: Histogram of the array 11 (slide19_098.gpr). Within array normalized data.

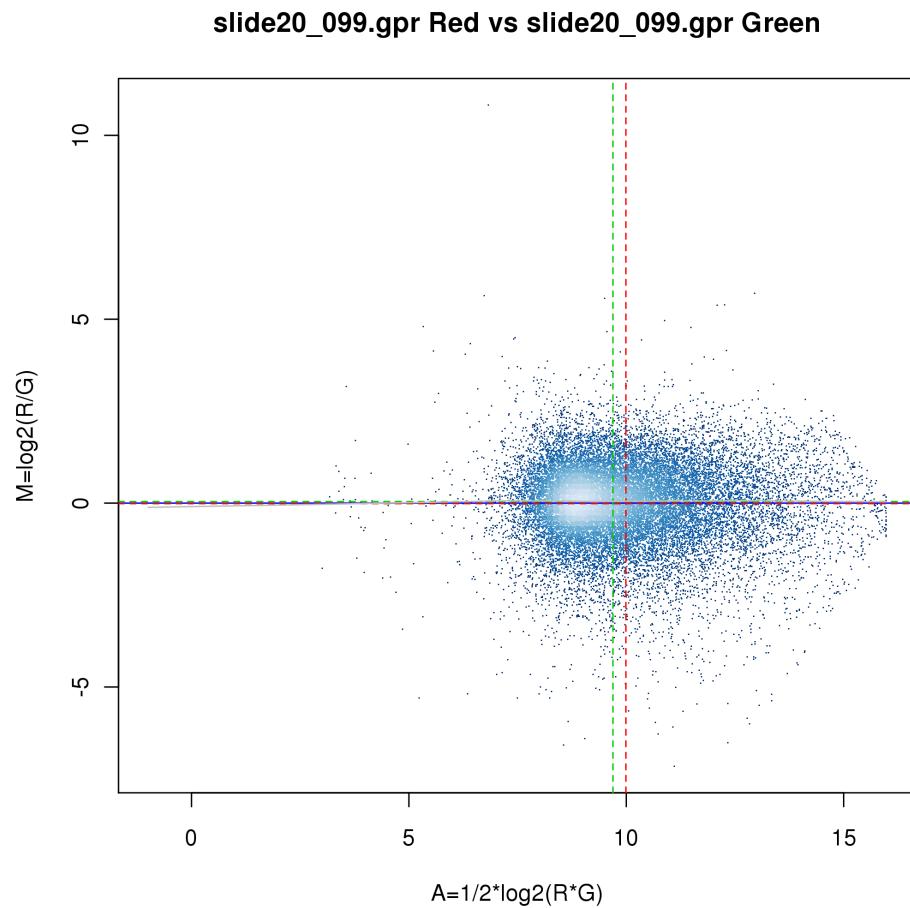


Figure 1.121: MA plot of array 12 (slide20_099.gpr). Within array normalized data.

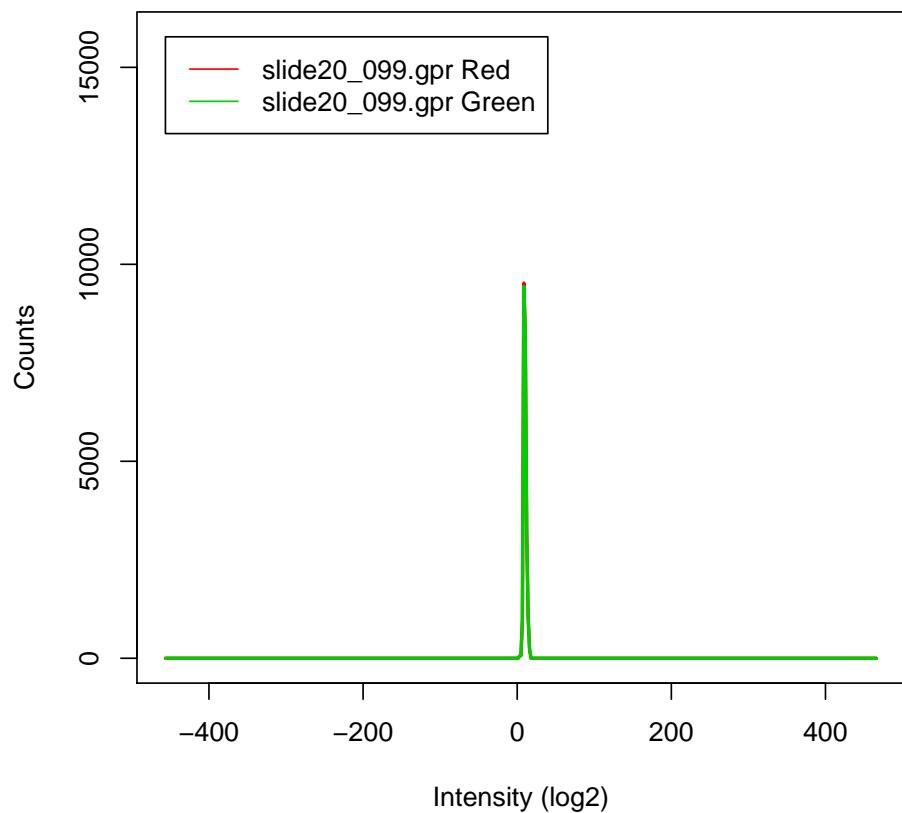


Figure 1.122: Histogram of the array 12 (slide20_099.gpr). Within array normalized data.

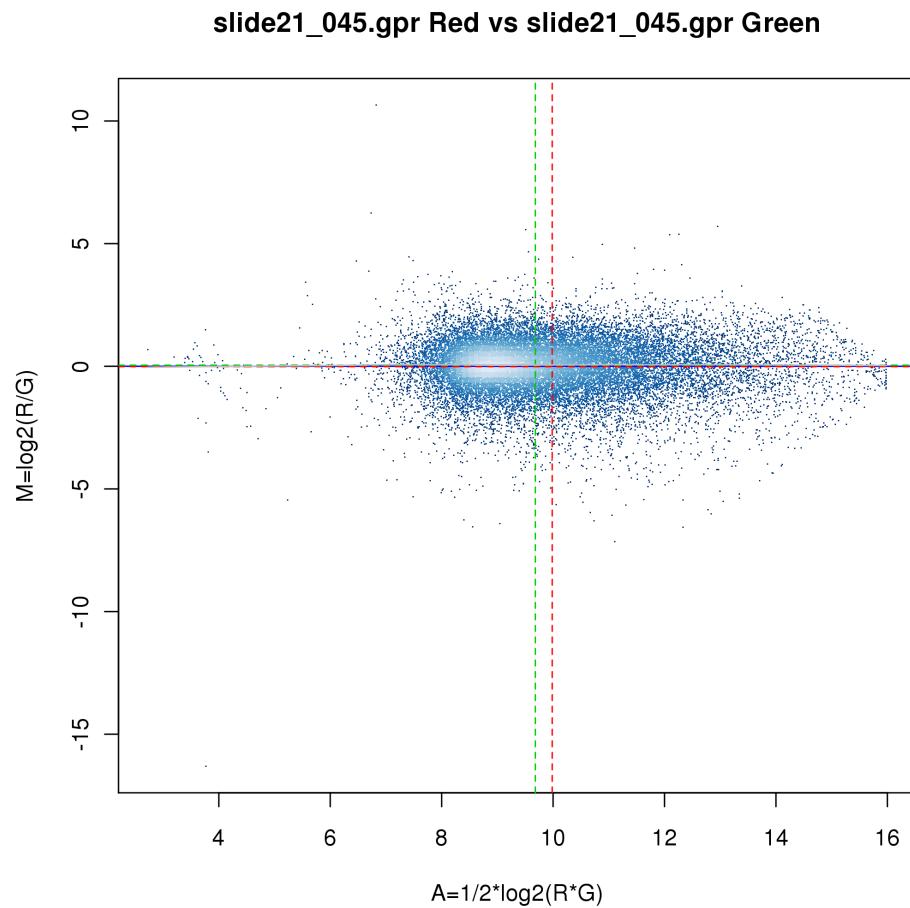


Figure 1.123: MA plot of array 13 (slide21_045.gpr). Within array normalized data.

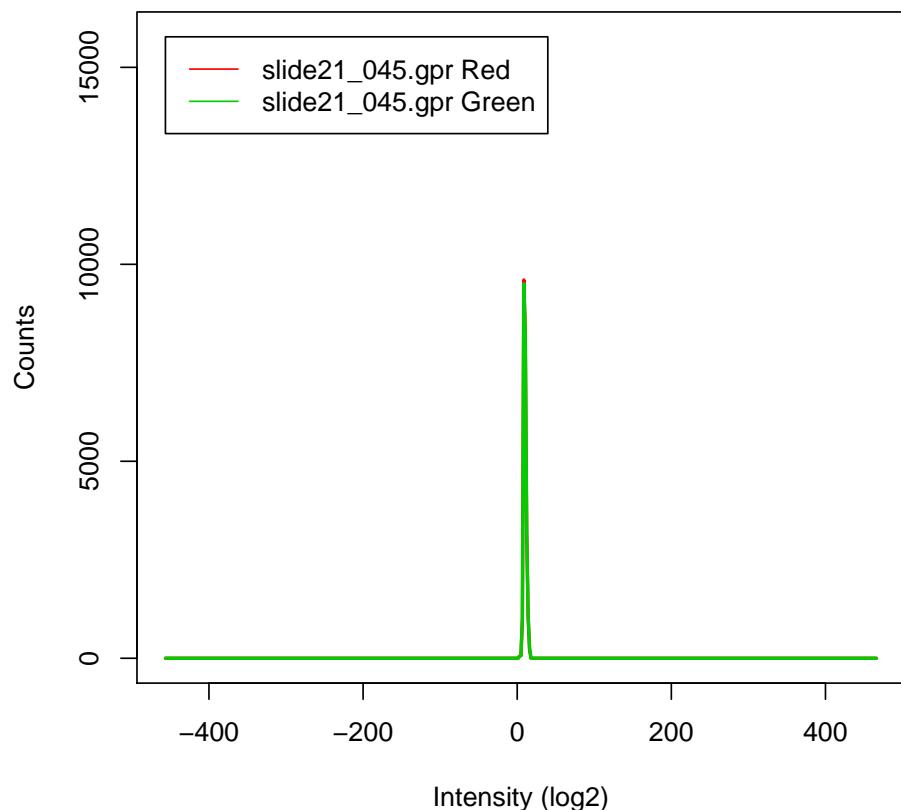


Figure 1.124: Histogram of the array 13 (slide21_045.gpr). Within array normalized data.

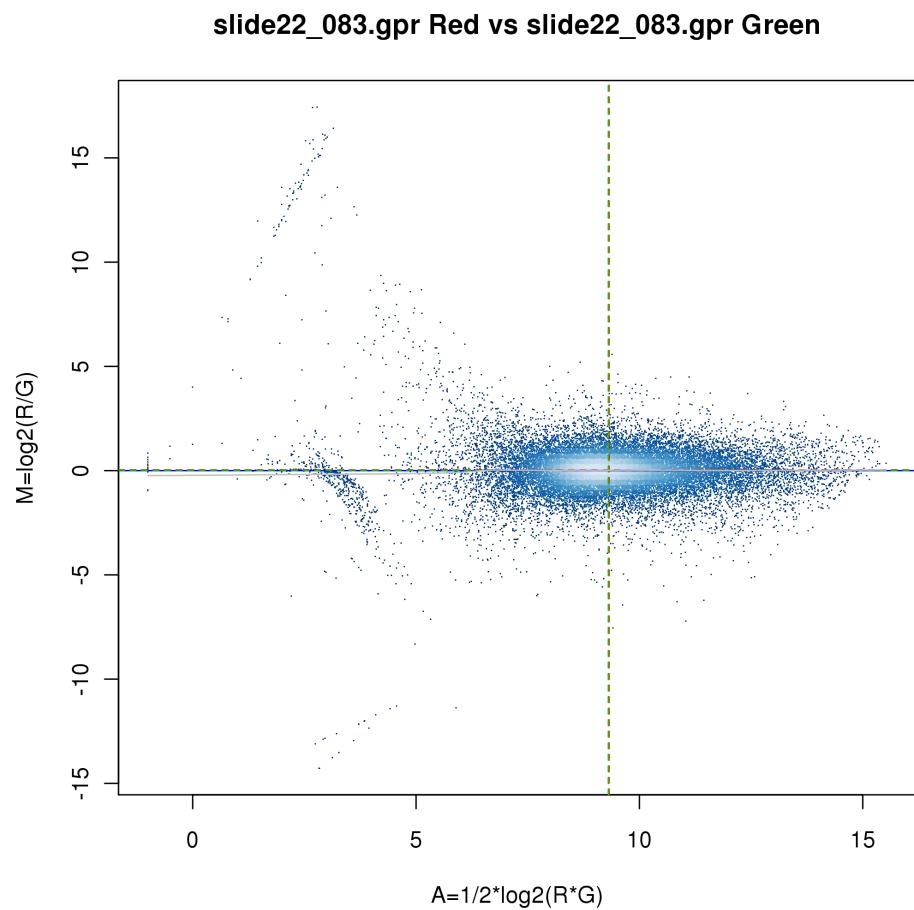


Figure 1.125: MA plot of array 14 (slide22_083.gpr). Within array normalized data.

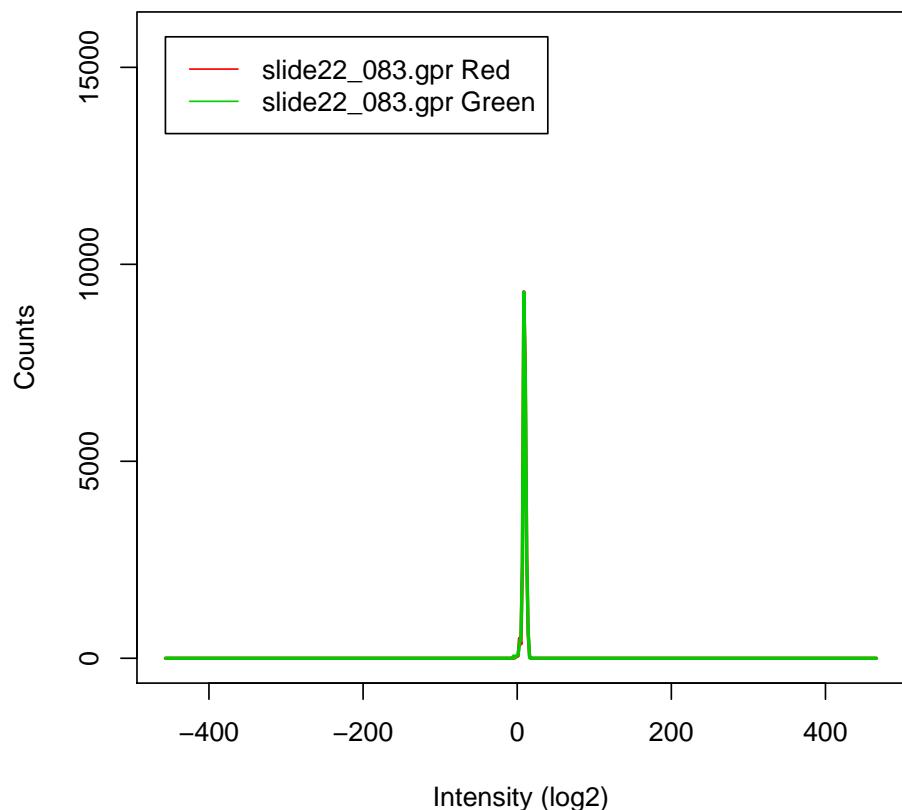


Figure 1.126: Histogram of the array 14 (slide22_083.gpr). Within array normalized data.

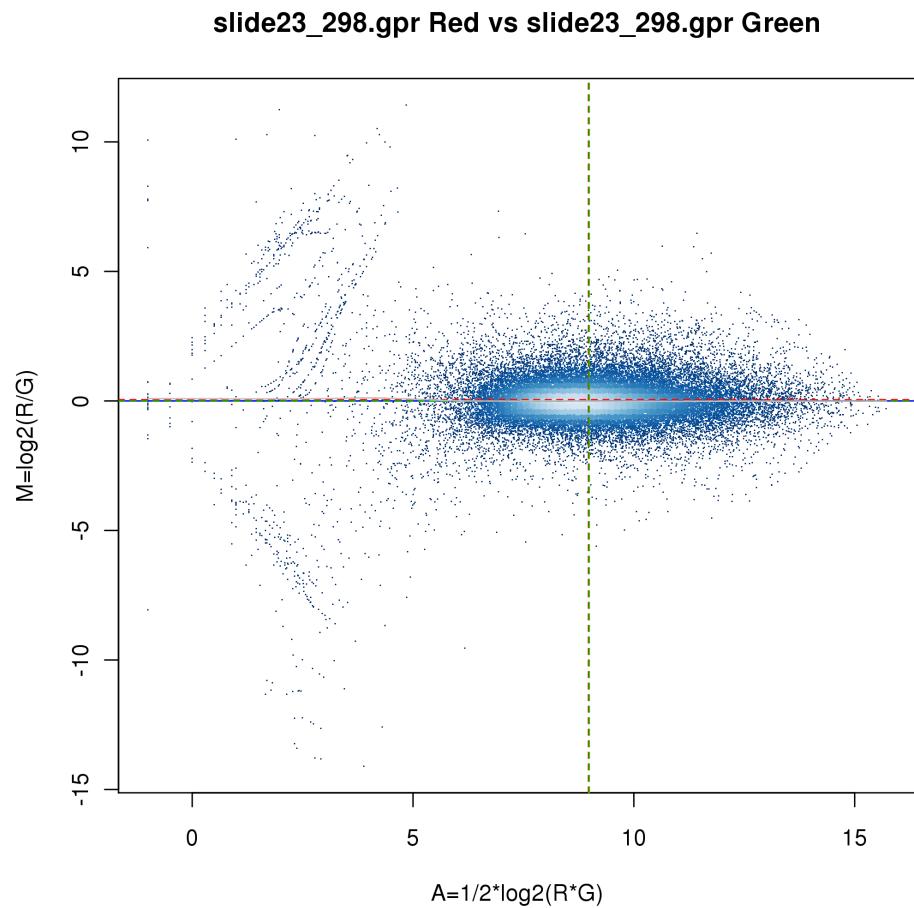


Figure 1.127: MA plot of array 15 (slide23_298.gpr). Within array normalized data.

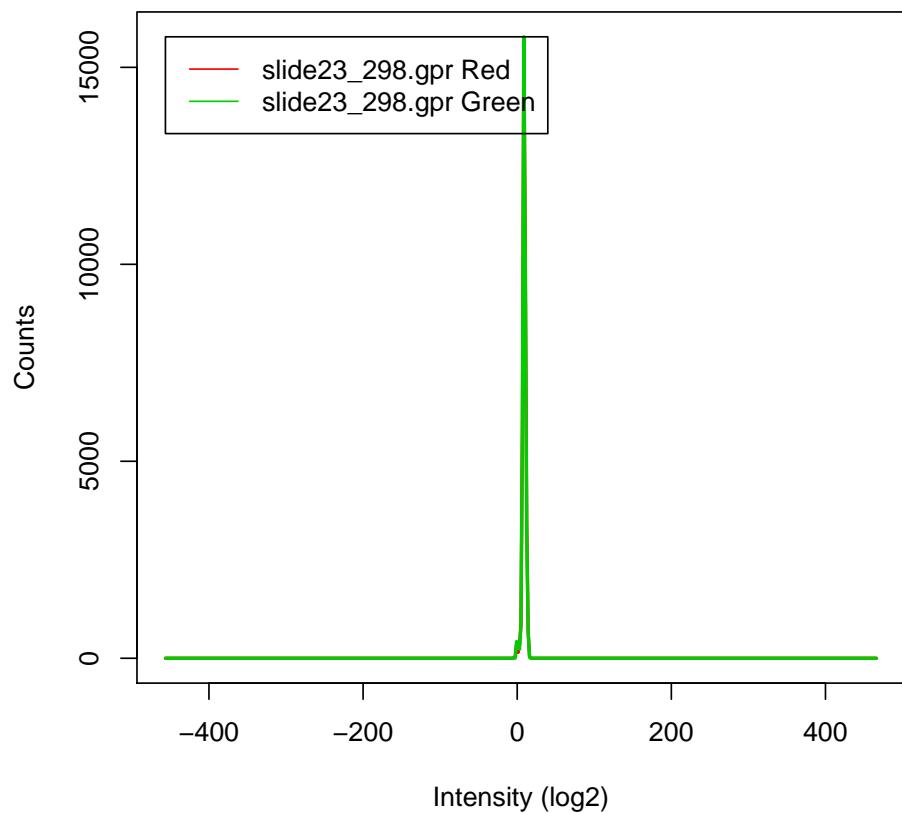


Figure 1.128: Histogram of the array 15 (slide23_298.gpr). Within array normalized data.

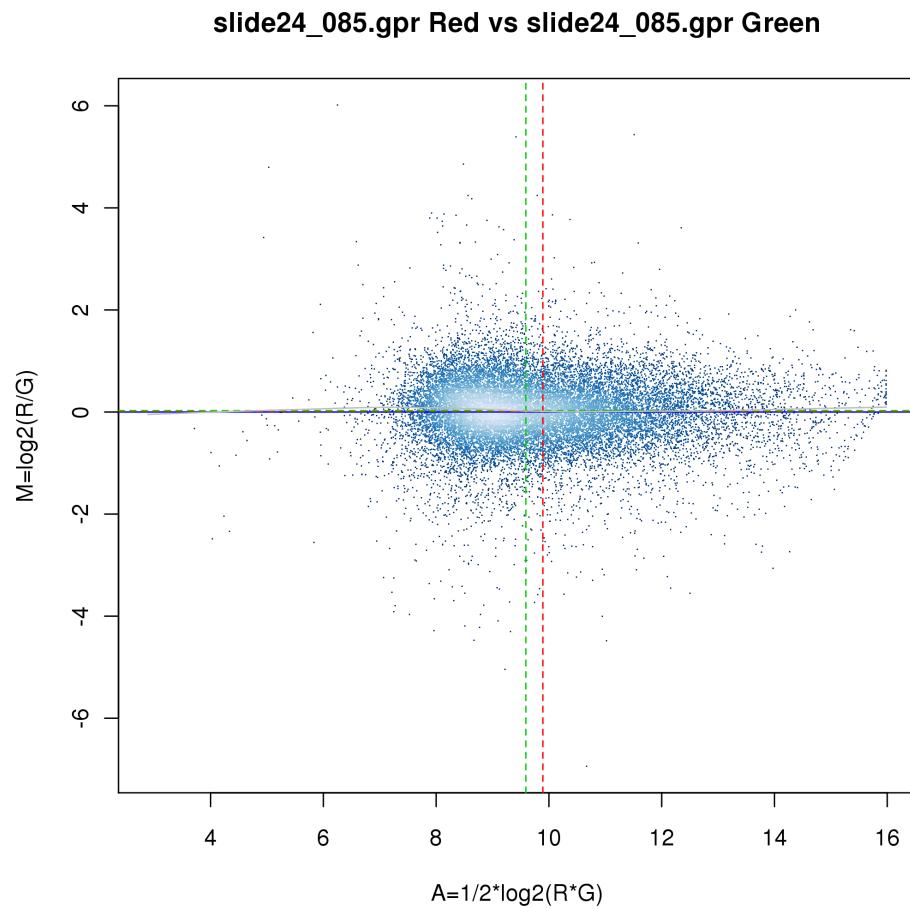


Figure 1.129: MA plot of array 16 (slide24_085.gpr). Within array normalized data.

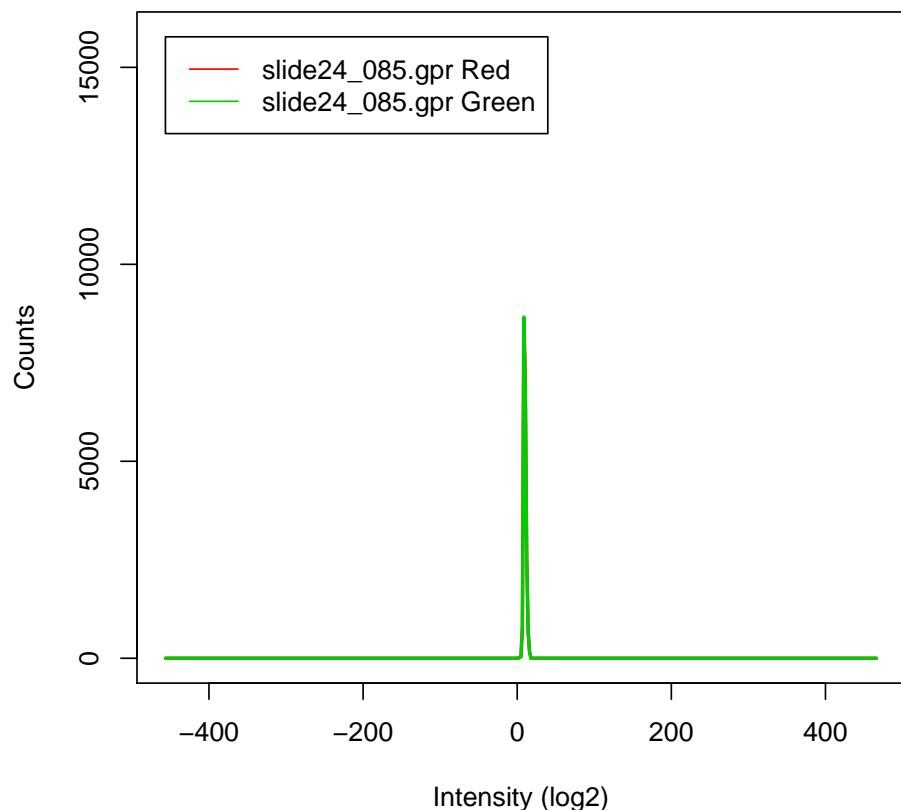


Figure 1.130: Histogram of the array 16 (slide24_085.gpr). Within array normalized data.

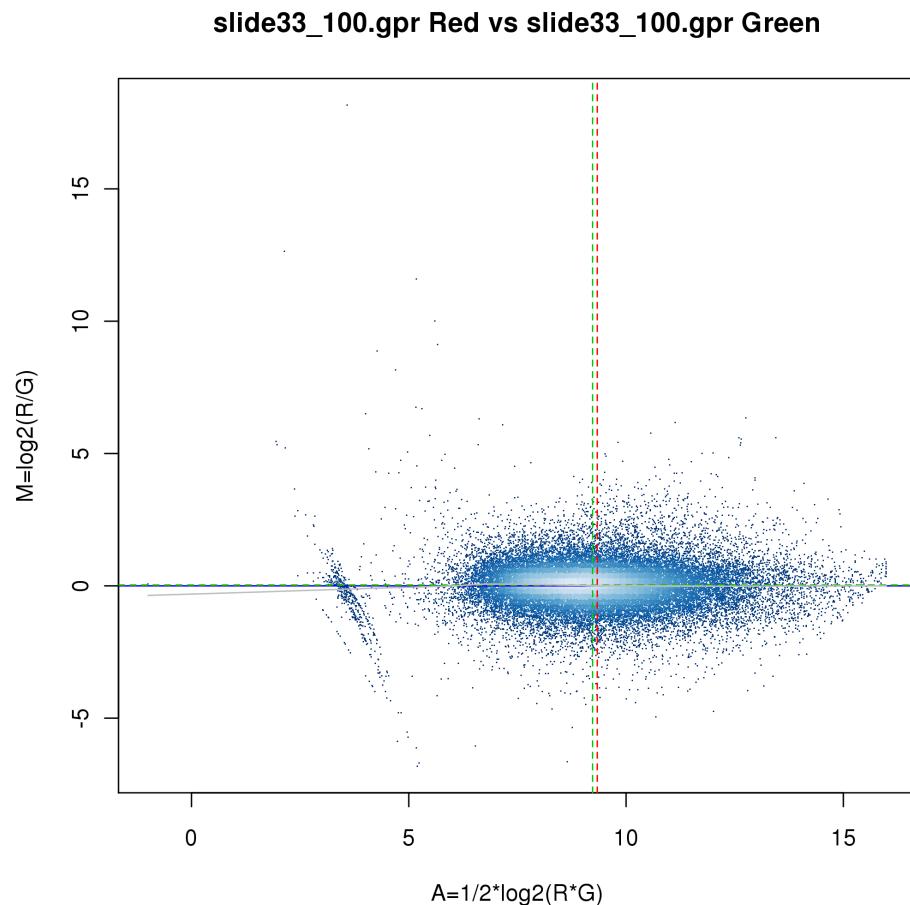


Figure 1.131: MA plot of array 17 (slide33_100.gpr). Within array normalized data.

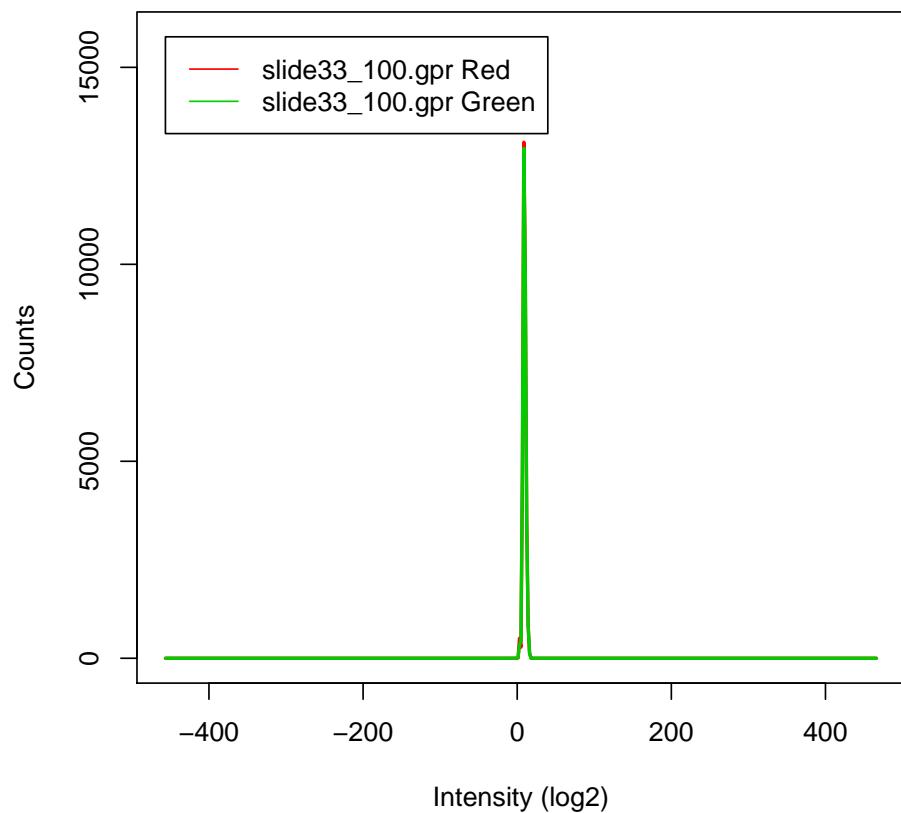


Figure 1.132: Histogram of the array 17 (slide33_100.gpr). Within array normalized data.

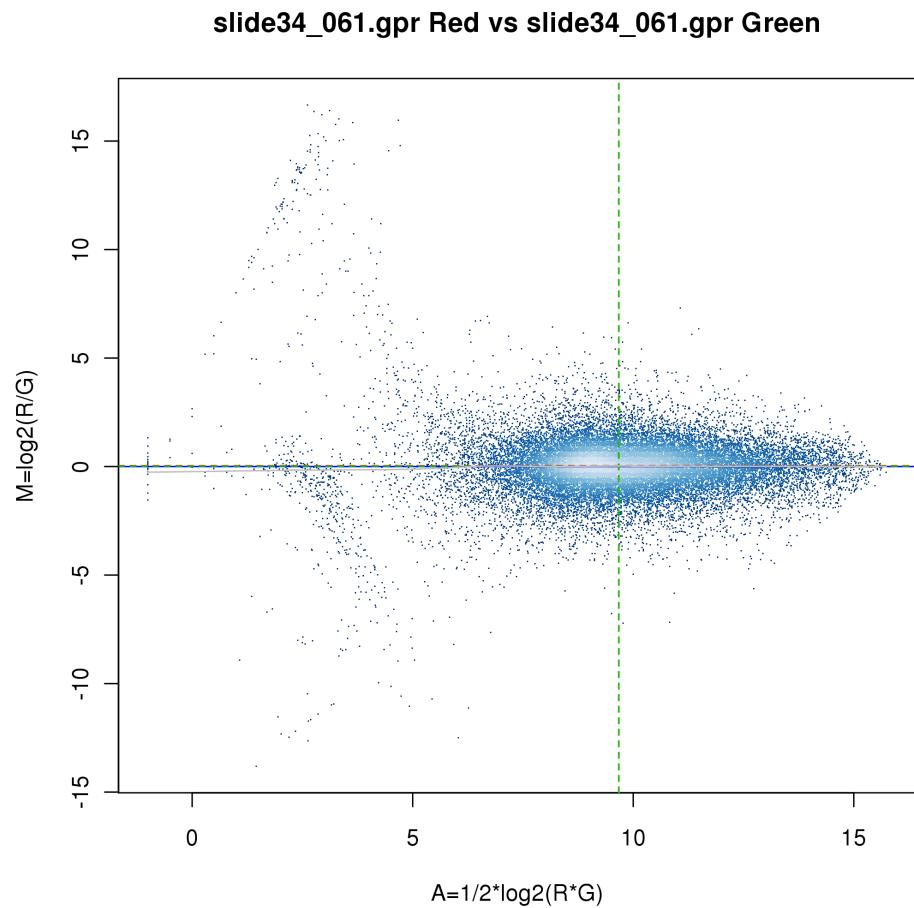


Figure 1.133: MA plot of array 18 (slide34_061.gpr). Within array normalized data.

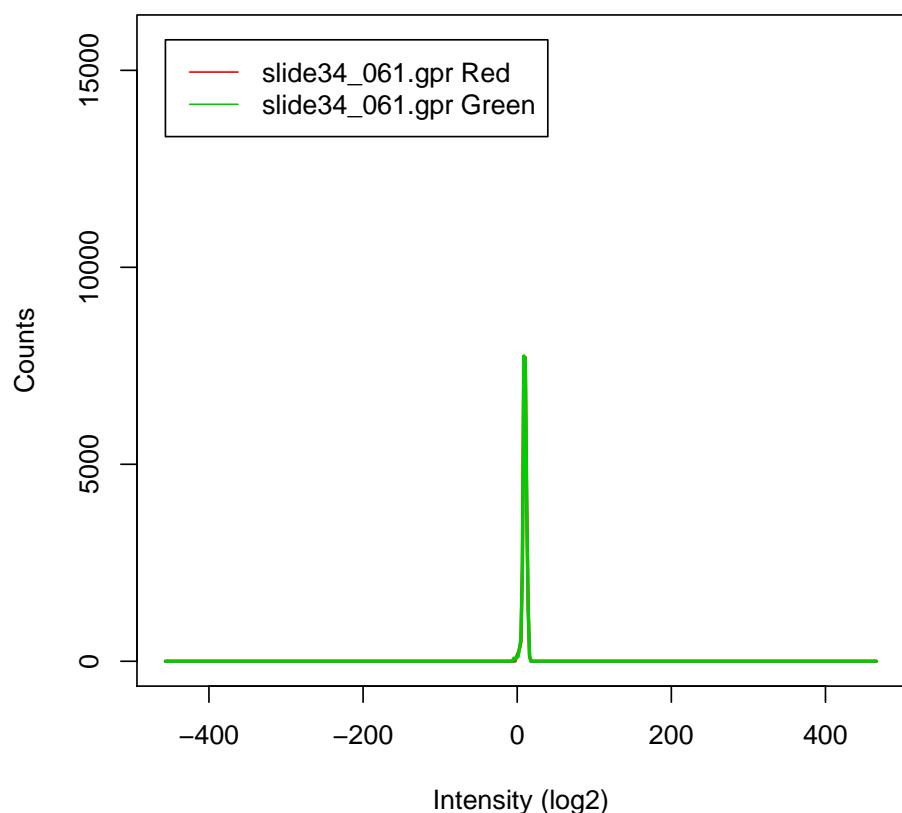


Figure 1.134: Histogram of the array 18 (slide34_061.gpr). Within array normalized data.

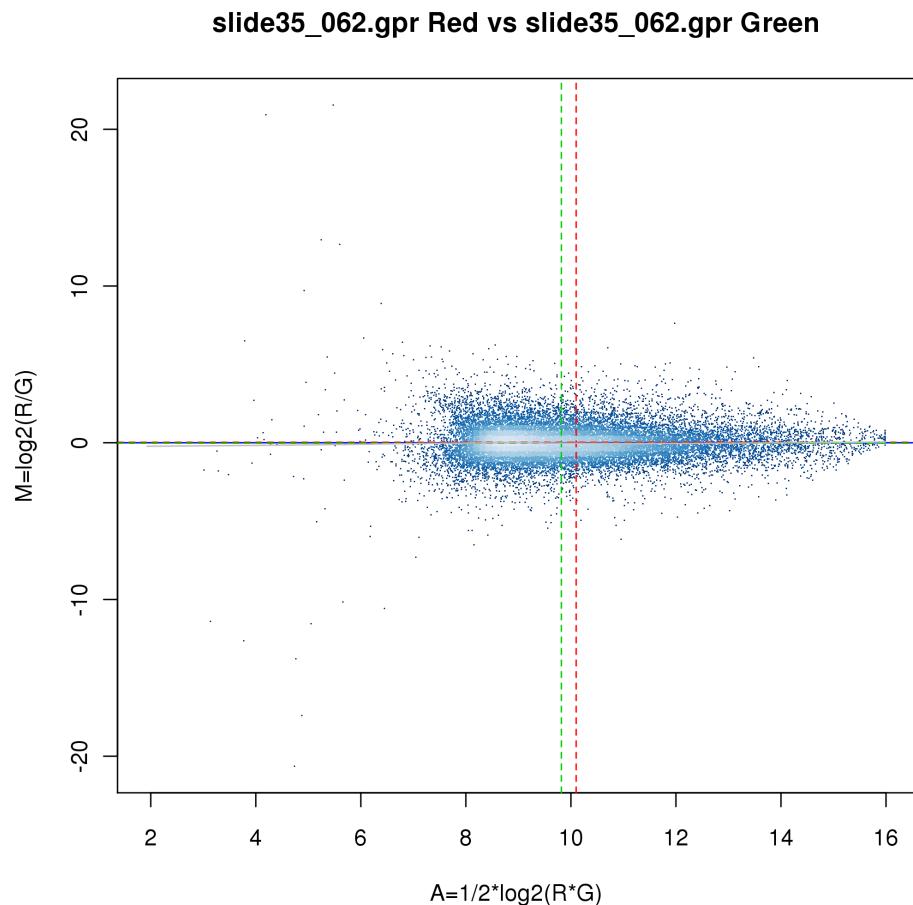


Figure 1.135: MA plot of array 19 (slide35_062.gpr). Within array normalized data.

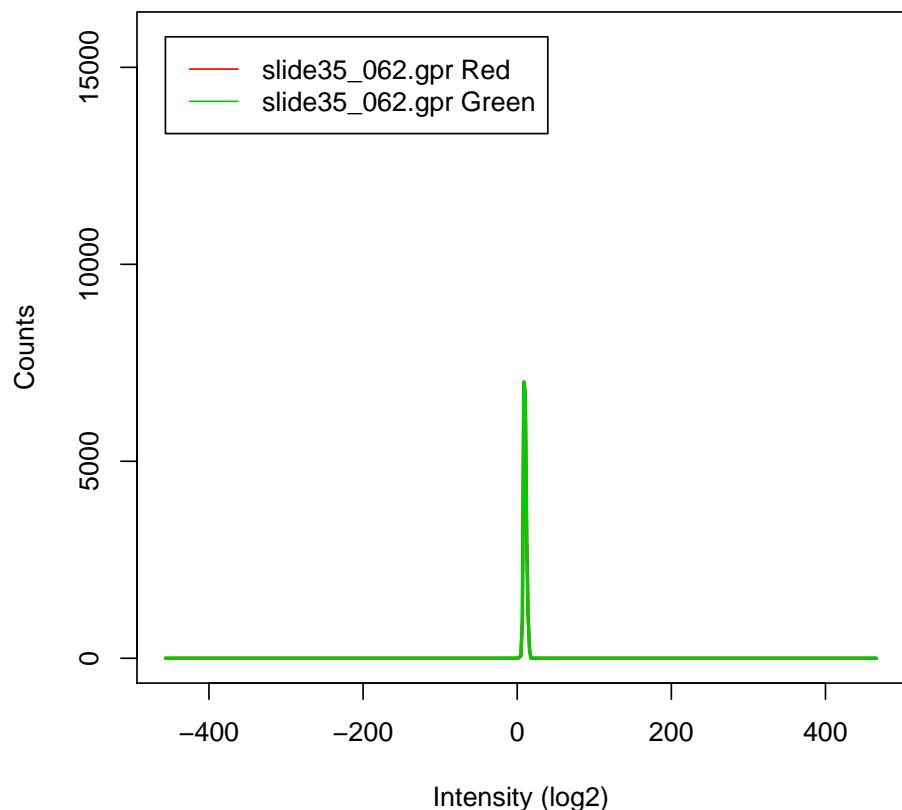


Figure 1.136: Histogram of the array 19 (slide35_062.gpr). Within array normalized data.

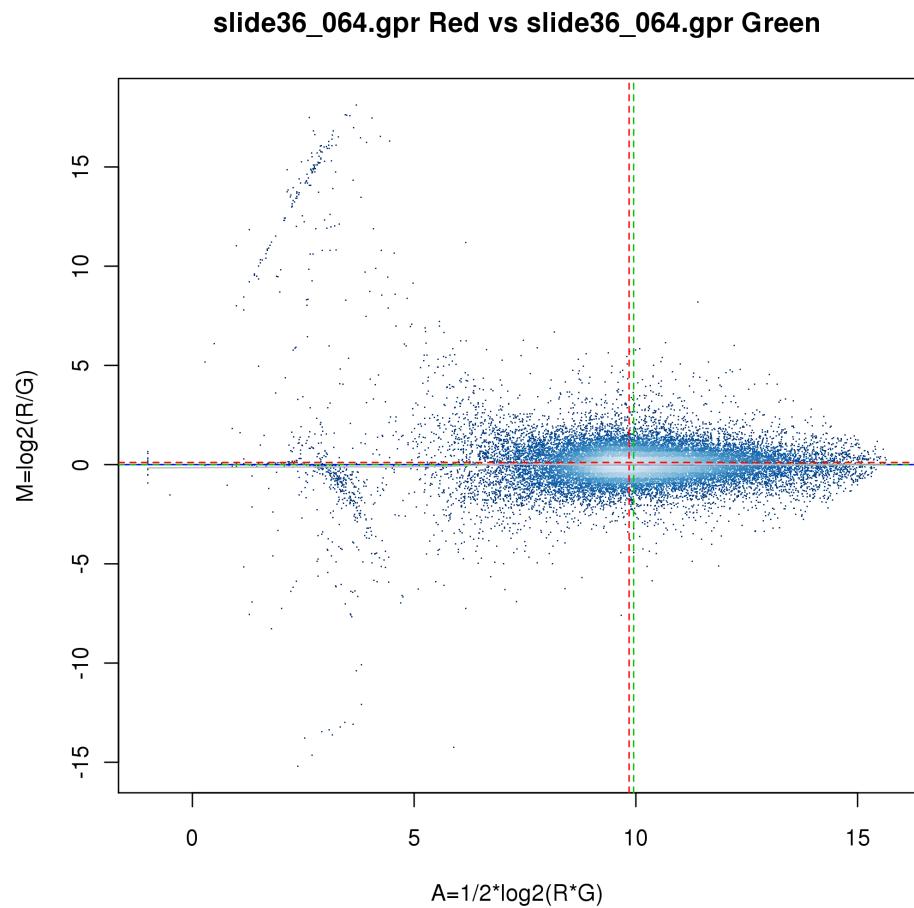


Figure 1.137: MA plot of array 20 (slide36_064.gpr). Within array normalized data.

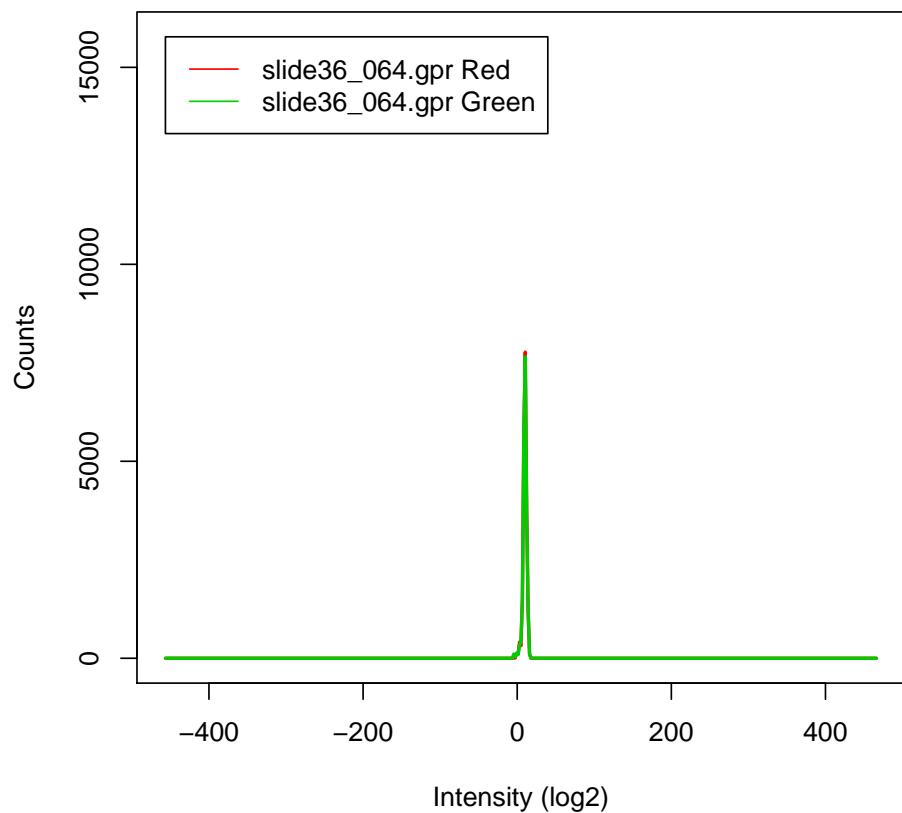


Figure 1.138: Histogram of the array 20 (slide36_064.gpr). Within array normalized data.

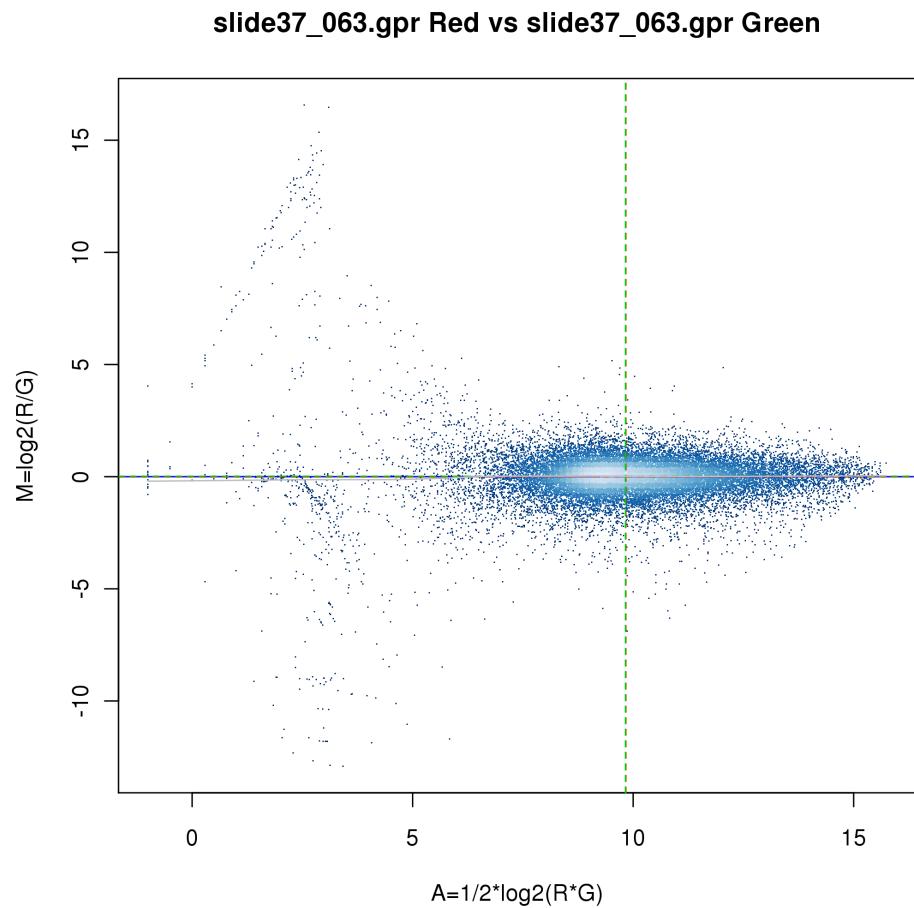


Figure 1.139: MA plot of array 21 (slide37_063.gpr). Within array normalized data.

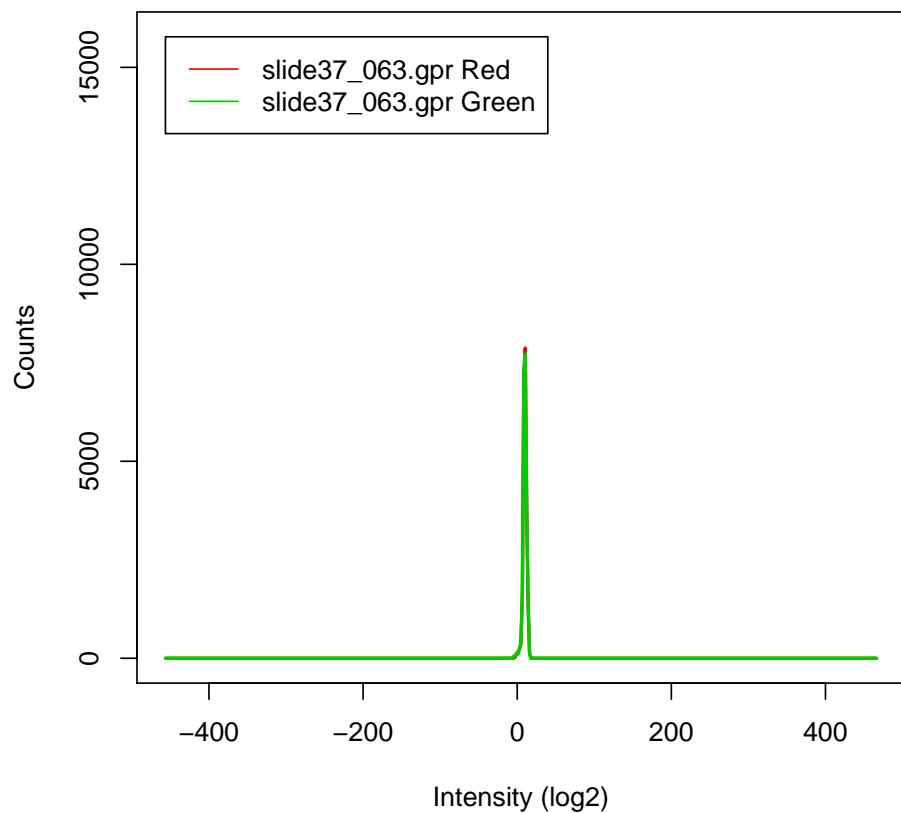


Figure 1.140: Histogram of the array 21 (slide37_063.gpr). Within array normalized data.

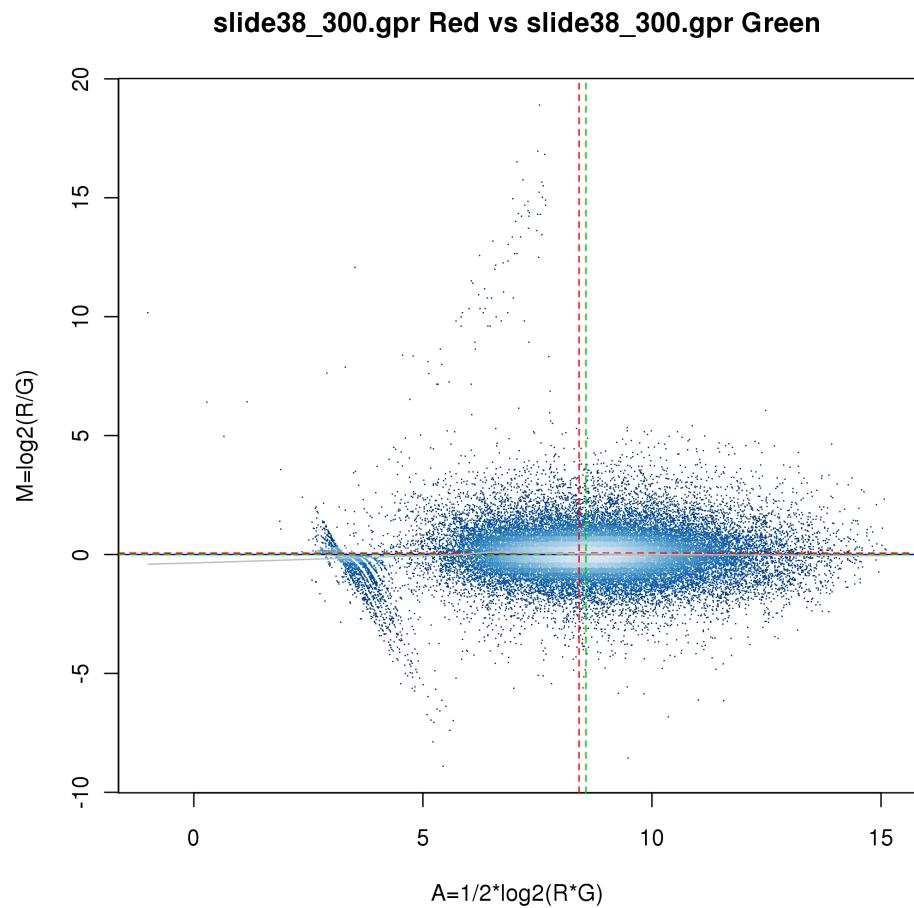


Figure 1.141: MA plot of array 22 (slide38_300.gpr). Within array normalized data.

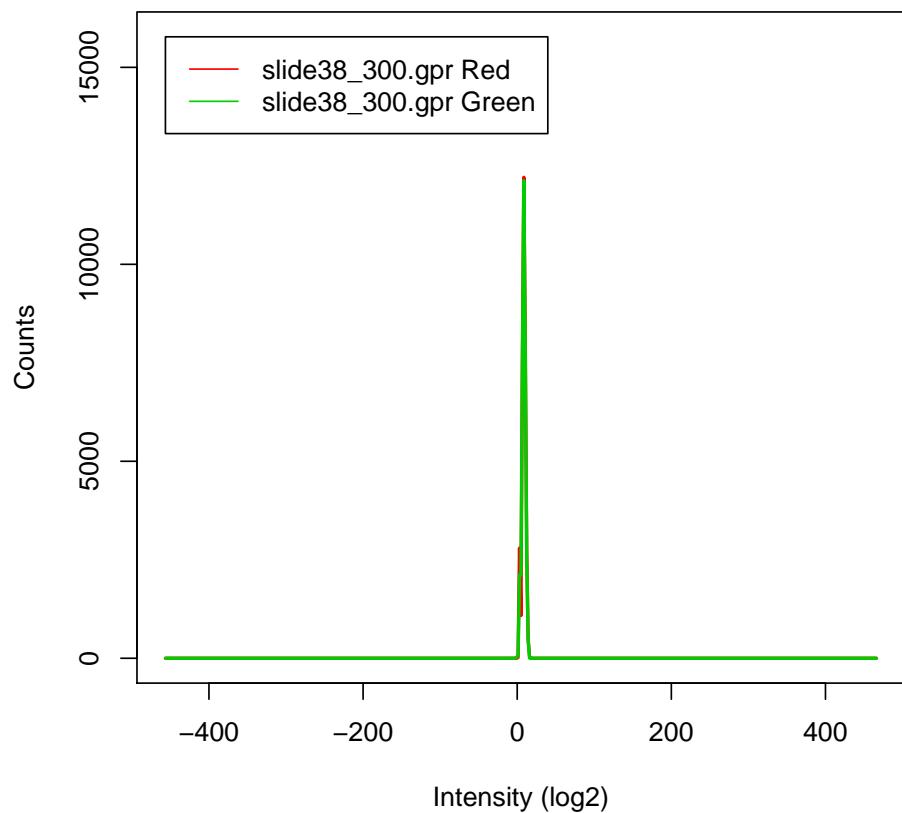


Figure 1.142: Histogram of the array 22 (slide38_300.gpr). Within array normalized data.

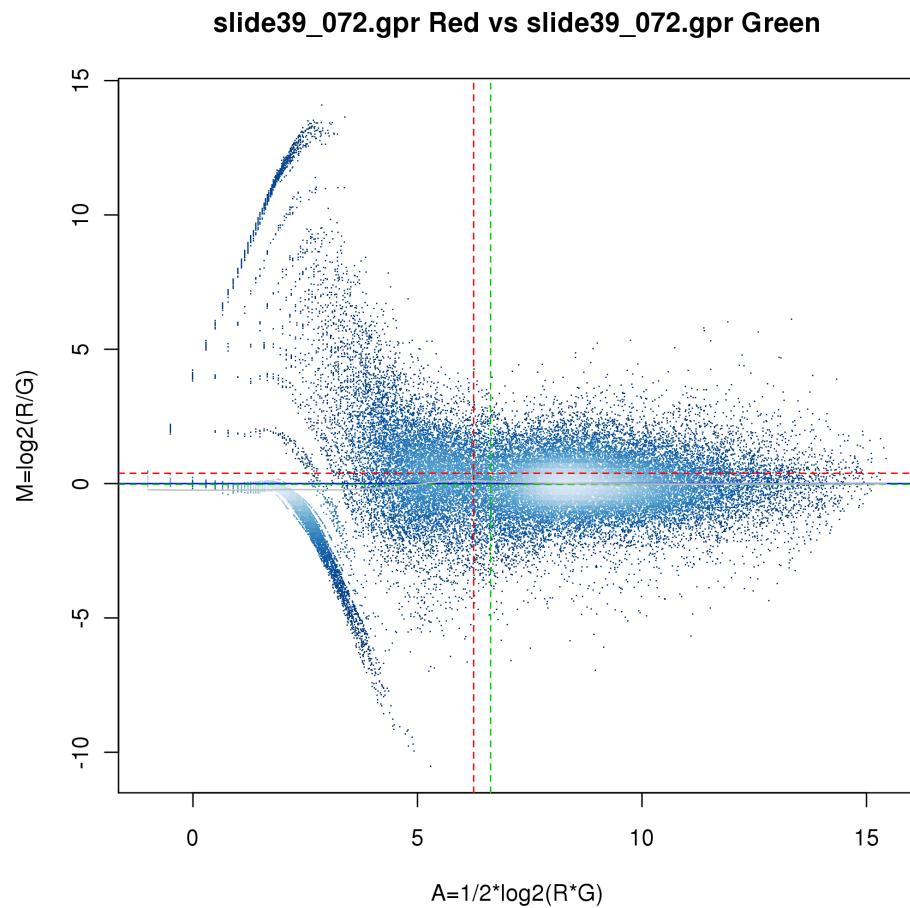


Figure 1.143: MA plot of array 23 (slide39_072.gpr). Within array normalized data.

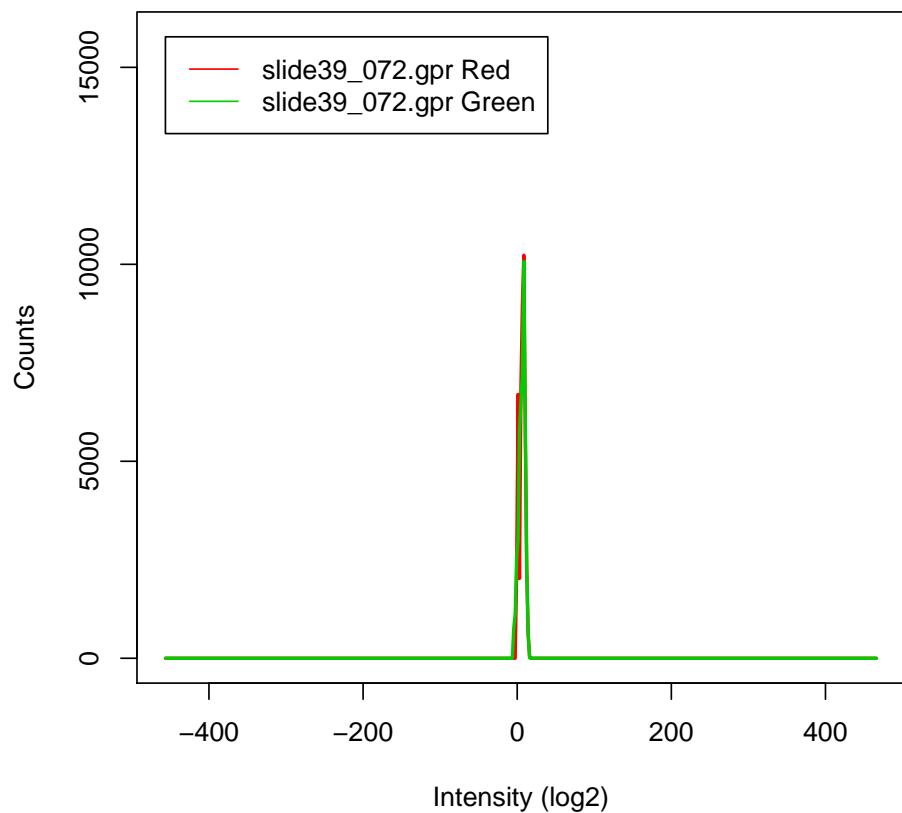


Figure 1.144: Histogram of the array 23 (slide39_072.gpr). Within array normalized data.

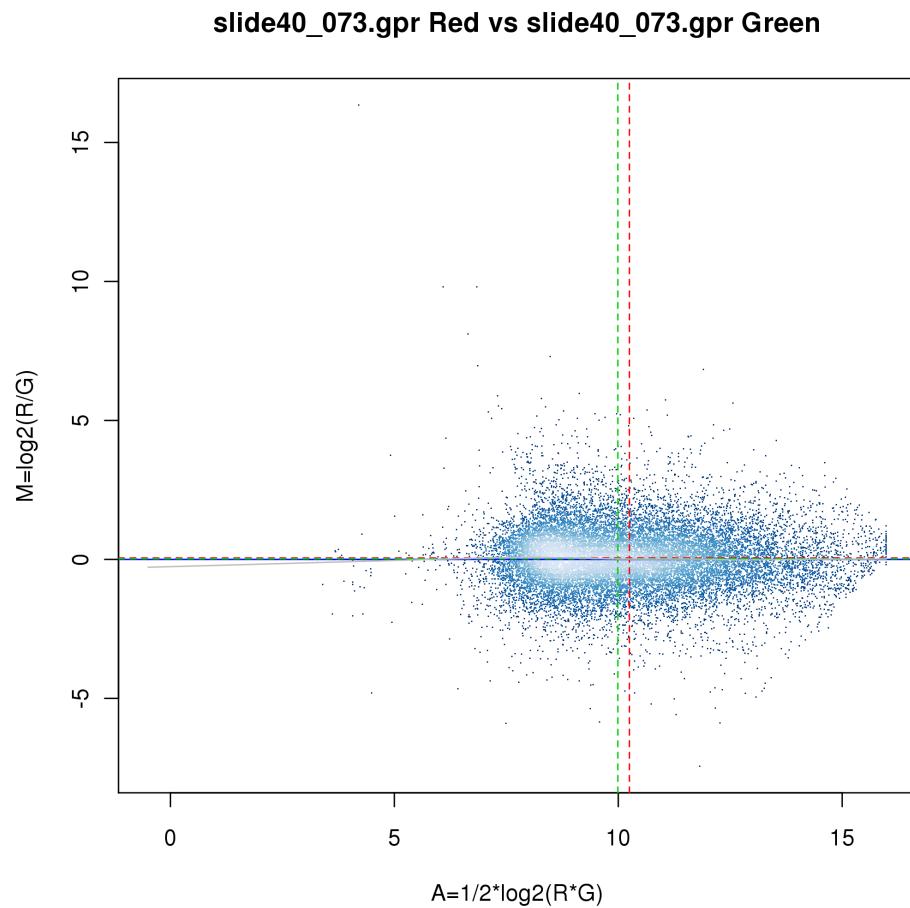


Figure 1.145: MA plot of array 24 (slide40_073.gpr). Within array normalized data.

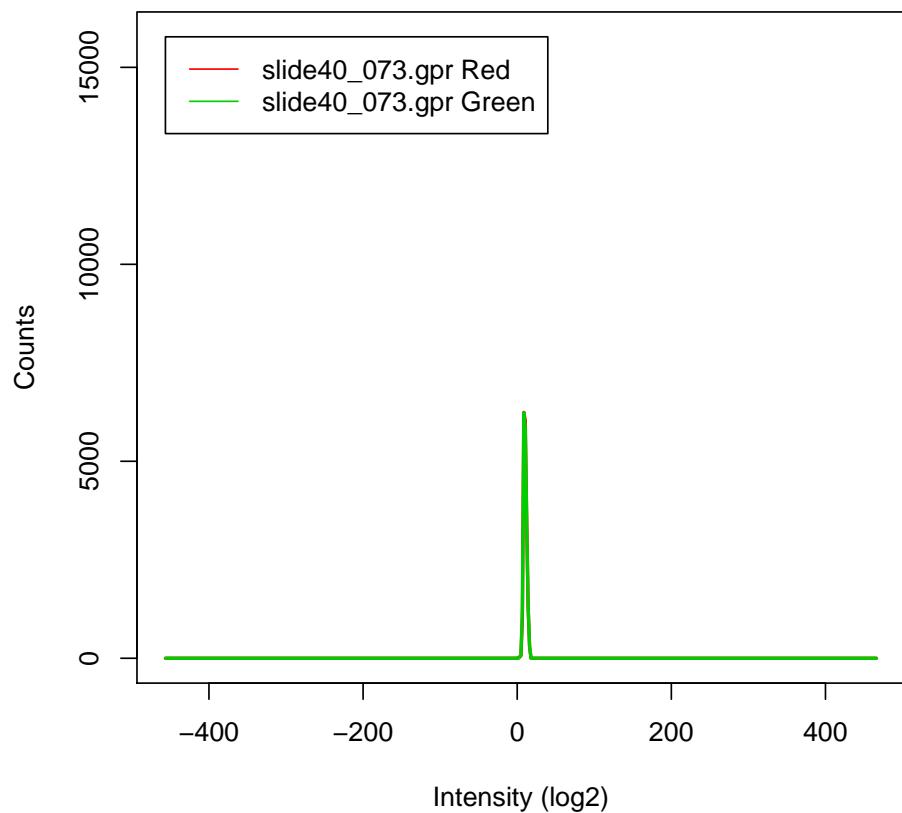


Figure 1.146: Histogram of the array 24 (slide40_073.gpr). Within array normalized data.

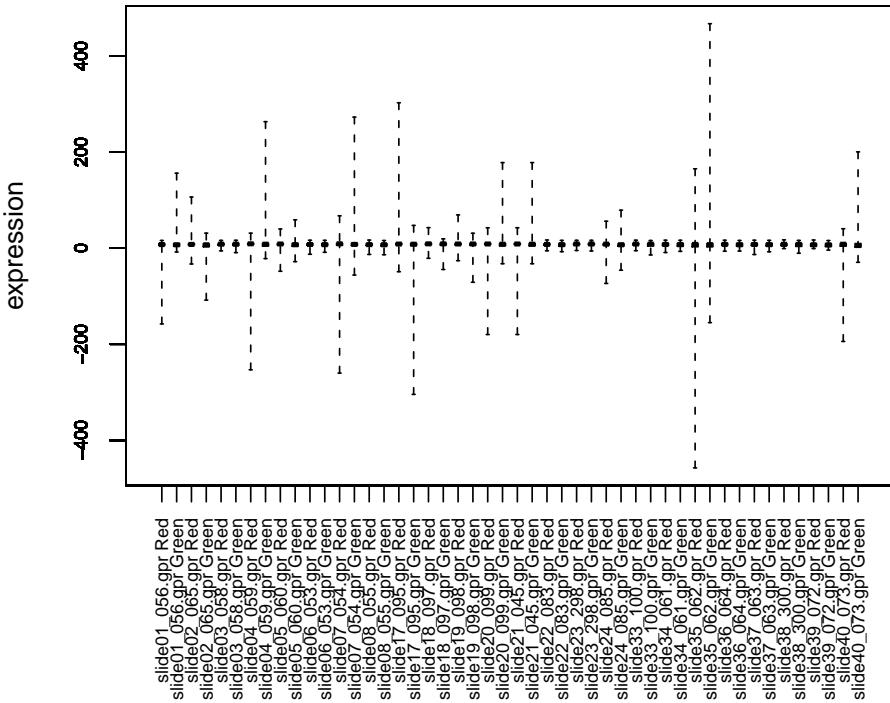


Figure 1.147: Boxplots of the signal intensities of each signal channel of the microarrays. Within array normalized data.

1.4 Between array normalization

The normalization of the expression values between the arrays in an micro array experiment adjusts the expression values (or log₂ regulation values), so that they have similar distribution over the arrays (and are therefore comparable). The histograms in the figures 1.148 and 1.198 show how the between-array-normalization has adjusted the normalized intensities across all arrays in this experiments.

```
> Dummy <- newMadbSet(Slides.norm)
```

```
Converting a limma MAList into a MadbSet...
```

```
Setting the weights... a weights of 0 means the gene was flagged, a weights of one means the signal is ok!
```

```

Inserting available annotation into the slot @genes

Inserting available annotation into the slot @genes

> drawHistogram(Dummy, lwd = 2, col = rep(c(2, 3), 24))

calculating histograms

```

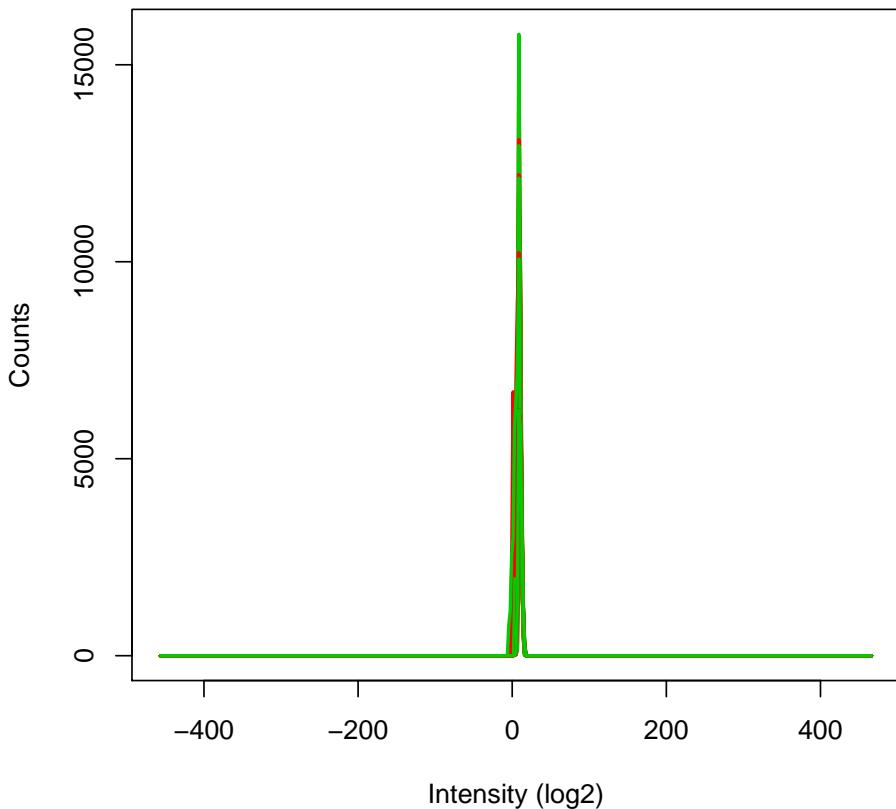


Figure 1.148: Histogram of all arrays within this experiment before the between-array-normalization. The green lines corresponds to the green signal channels and the red line to the red channel respectively.

```
> Slides.norm <- normalizeBetweenArrays(Slides.norm, method = "scale")
```

Next diagnostic plots of the (between array) normalized data will be drawn.

```
> Dummy <- newMadbSet(Slides.norm)
```

Converting a limma MAList into a MadbSet...

Setting the weights... a weights of 0 means the gene was flagged, a weights of one means the signal is ok!

```
Inserting available annotation into the slot @genes
```

```
Inserting available annotation into the slot @genes
```

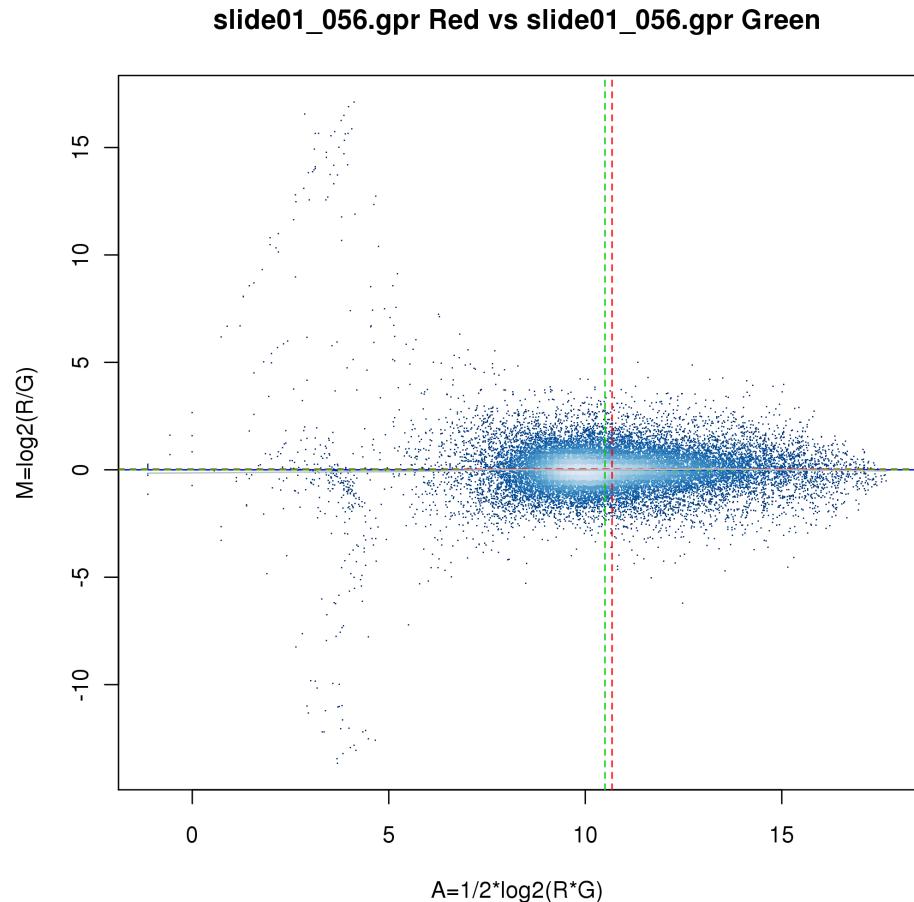


Figure 1.149: MA plot of array 1 (slide01_056.gpr). Between array normalized data.

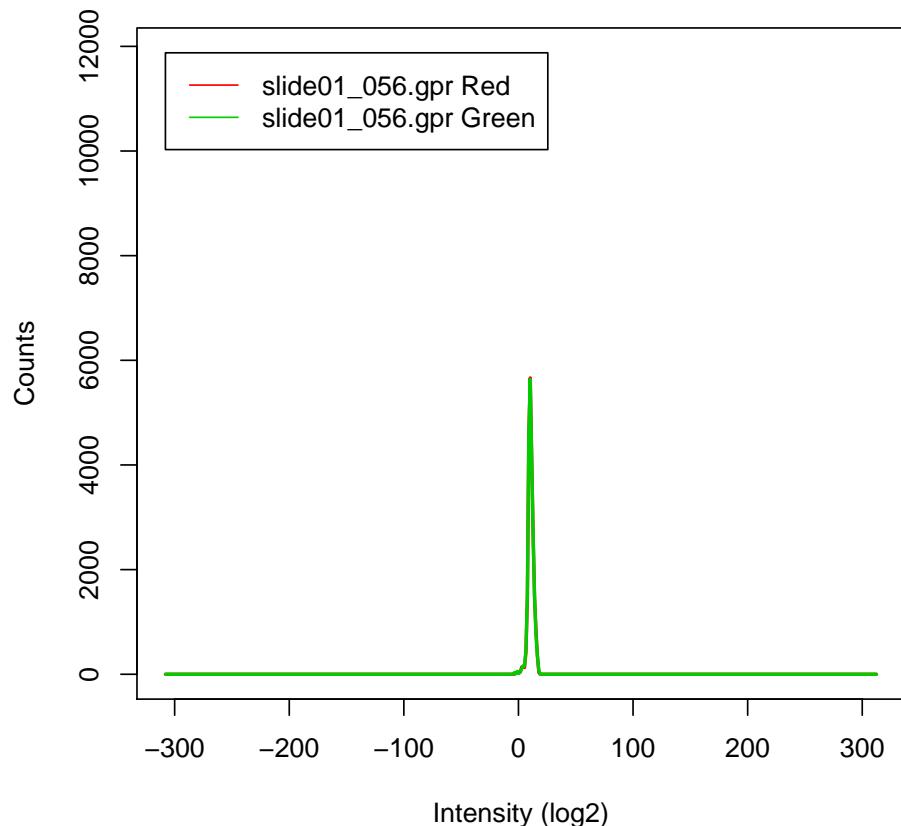


Figure 1.150: Histogram of the array 1 (slide01_056.gpr). Between array normalized data.

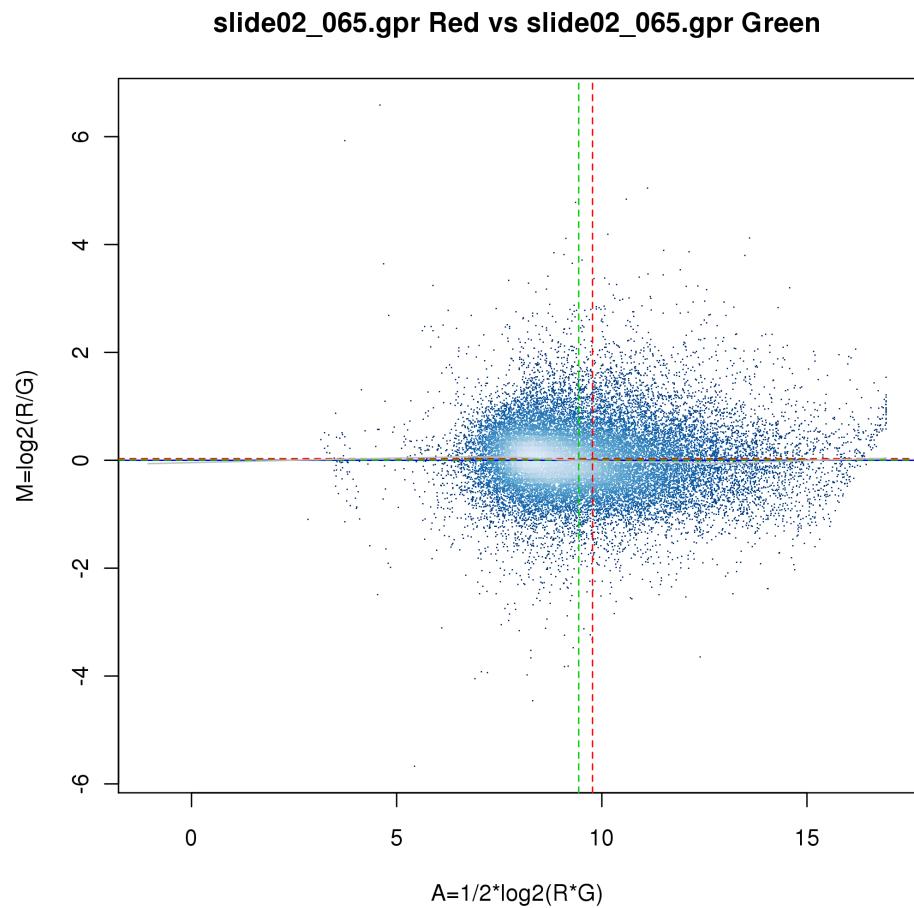


Figure 1.151: MA plot of array 2 (slide02_065.gpr). Between array normalized data.

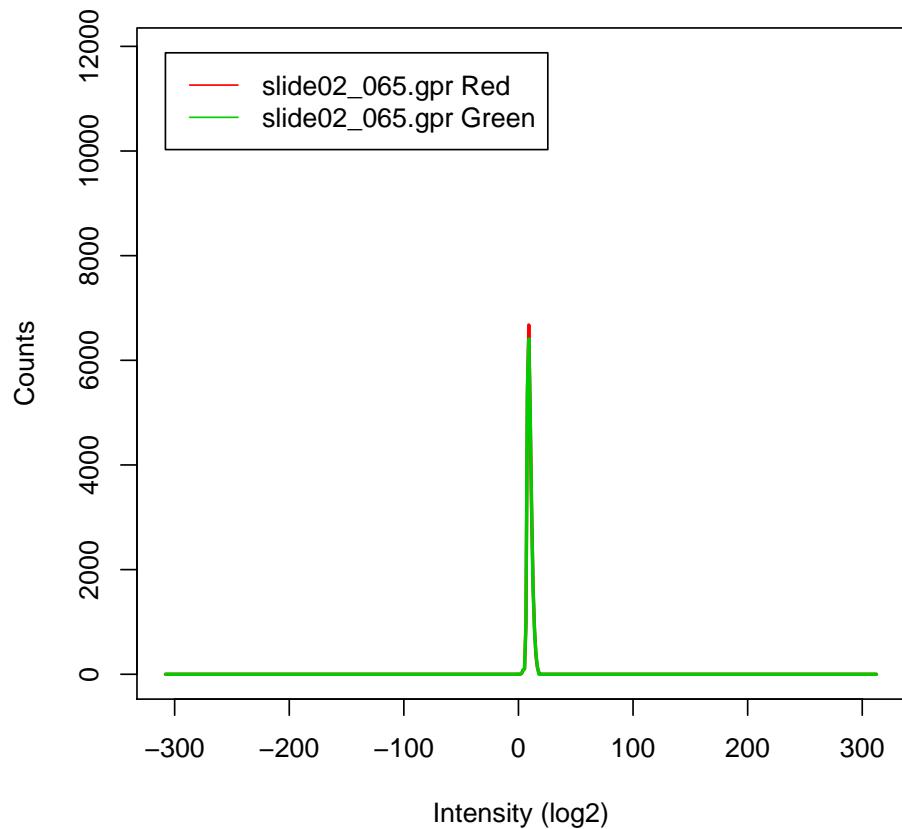


Figure 1.152: Histogram of the array 2 (slide02_065.gpr). Between array normalized data.

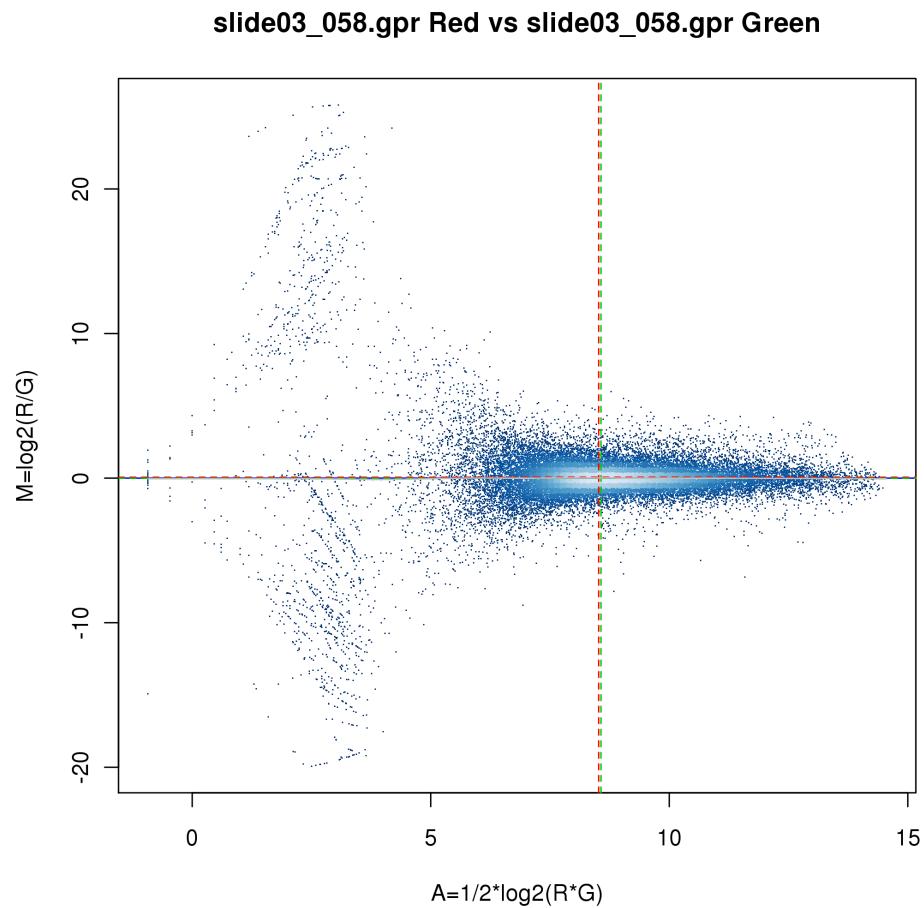


Figure 1.153: MA plot of array 3 (slide03_058.gpr). Between array normalized data.

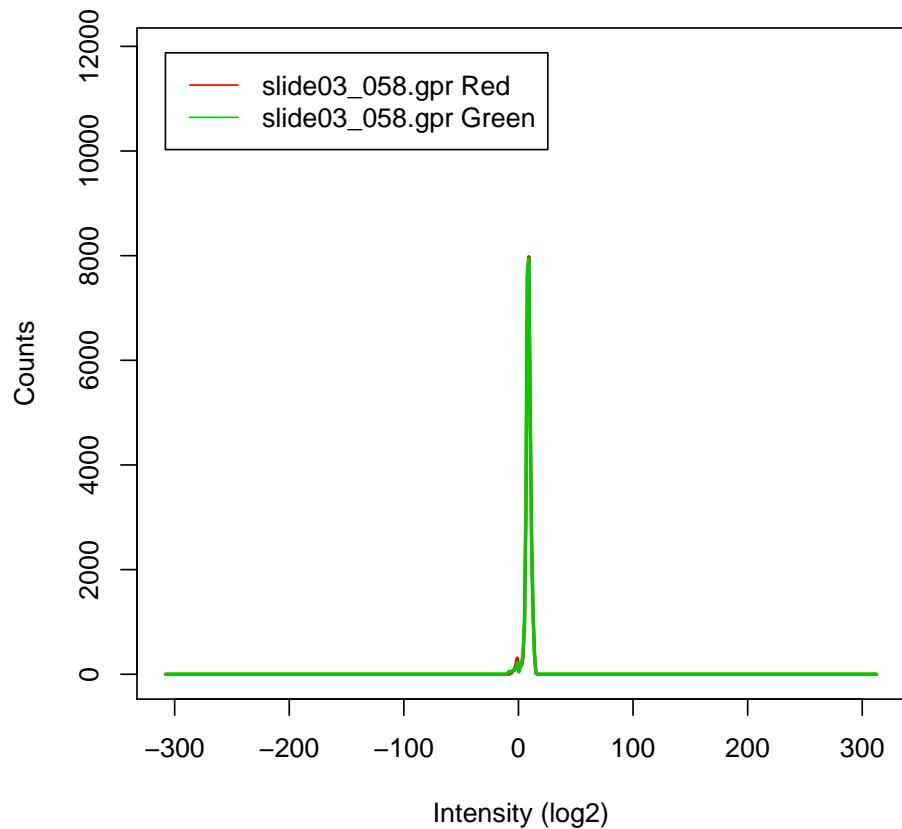


Figure 1.154: Histogram of the array 3 (slide03_058.gpr). Between array normalized data.

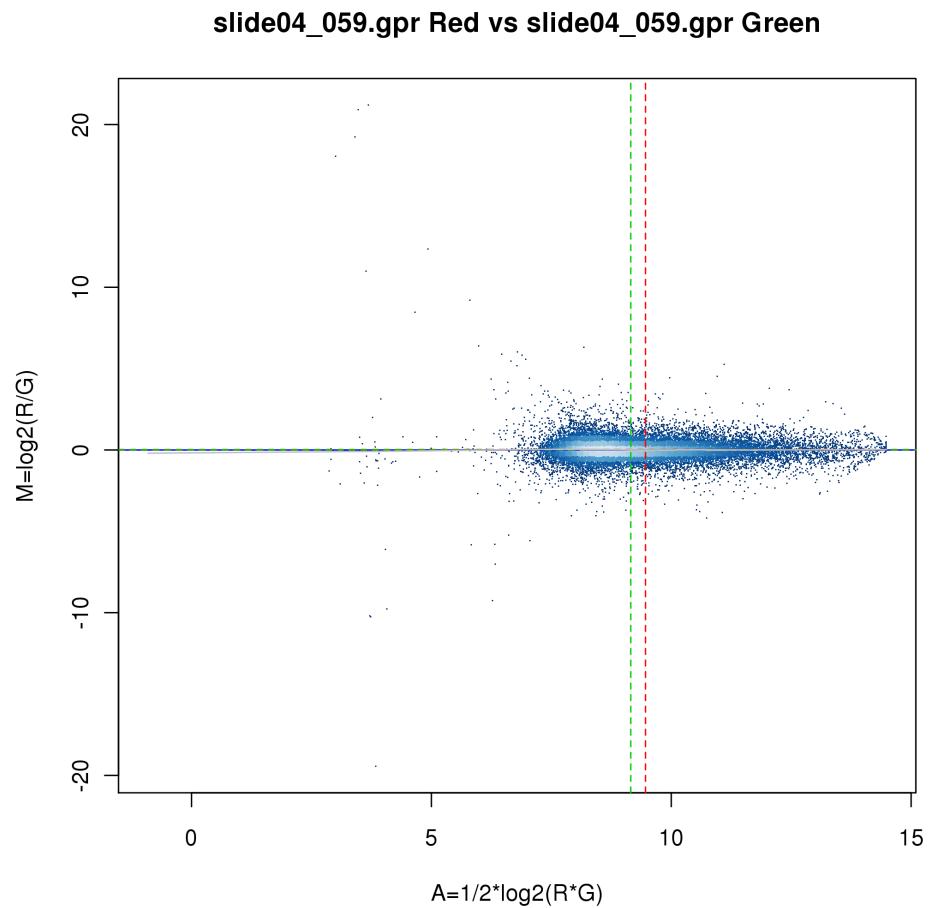


Figure 1.155: MA plot of array 4 (slide04_059.gpr). Between array normalized data.

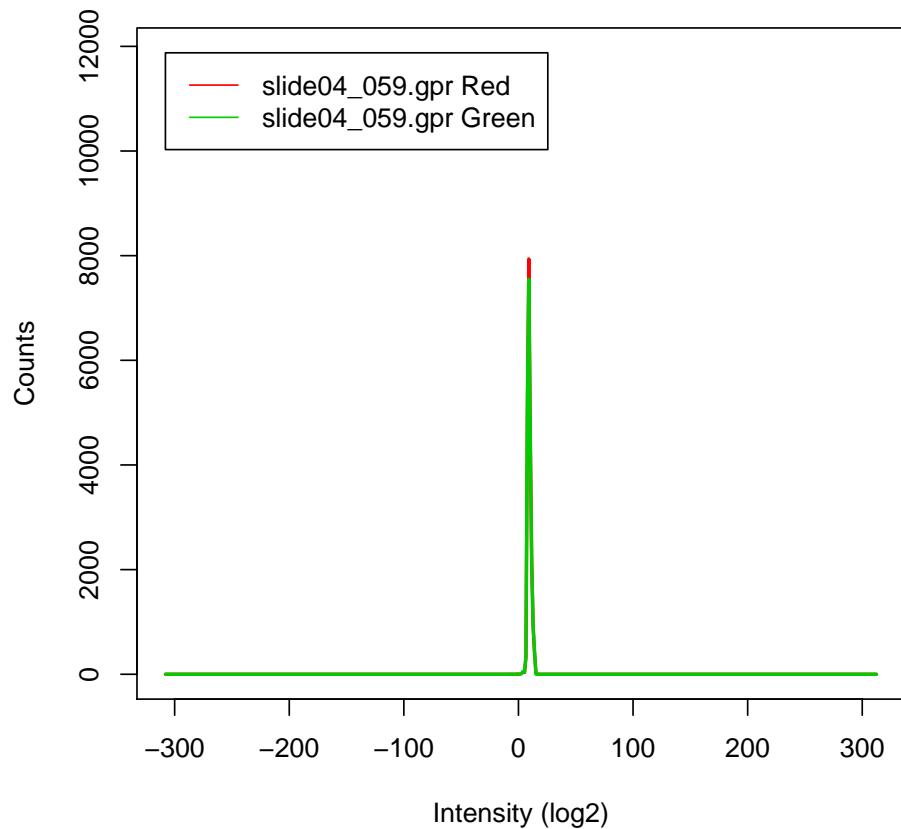


Figure 1.156: Histogram of the array 4 (slide04_059.gpr). Between array normalized data.

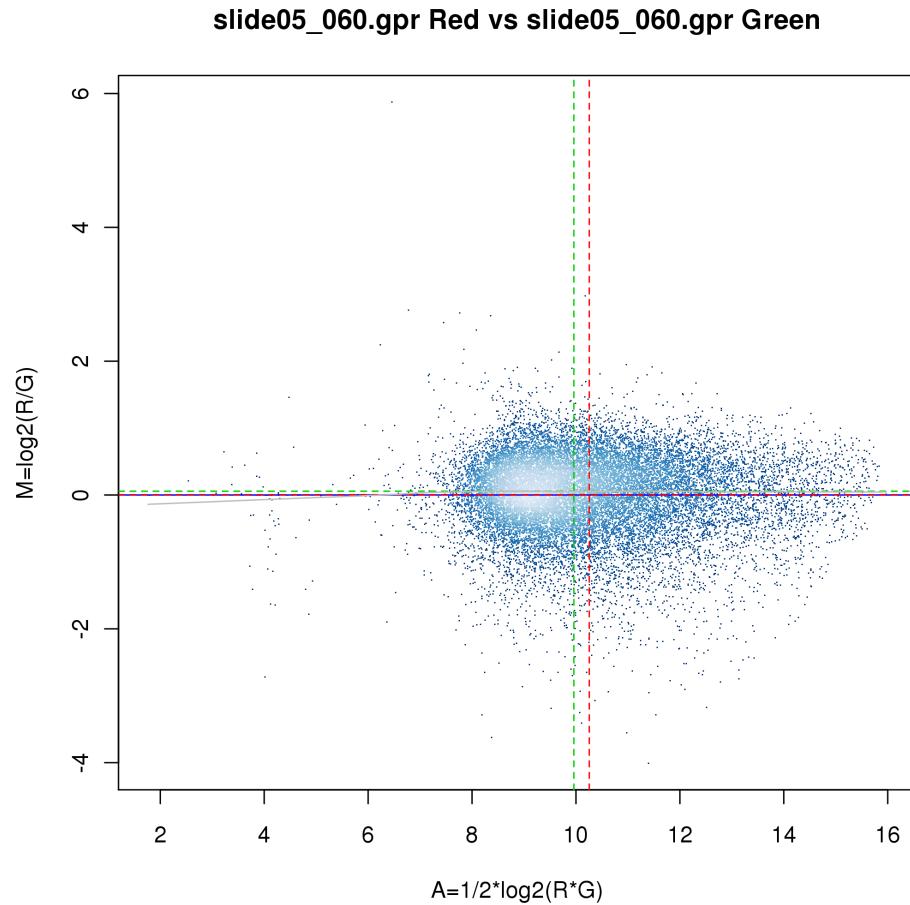


Figure 1.157: MA plot of array 5 (slide05_060.gpr). Between array normalized data.

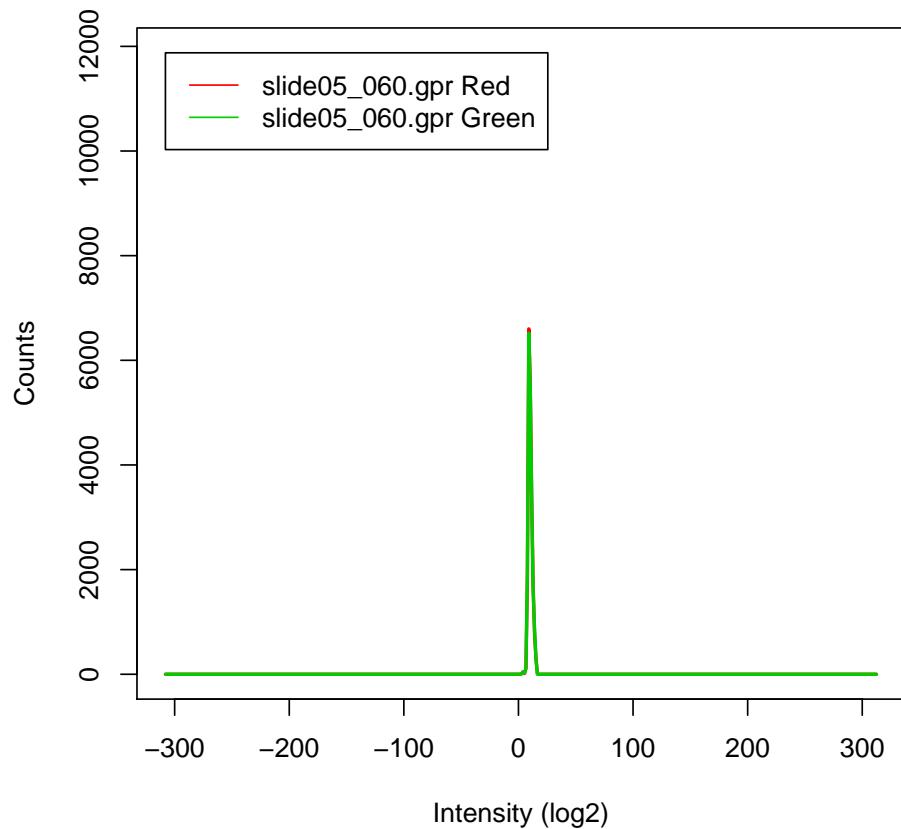


Figure 1.158: Histogram of the array 5 (slide05_060.gpr). Between array normalized data.

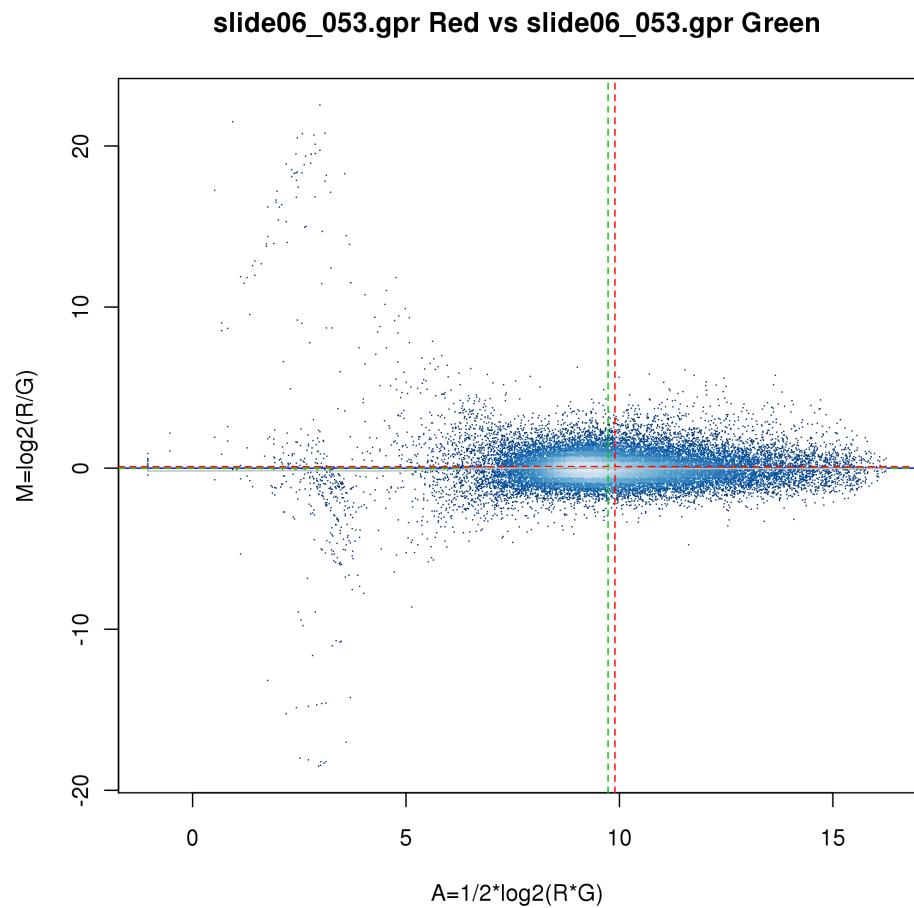


Figure 1.159: MA plot of array 6 (slide06_053.gpr). Between array normalized data.

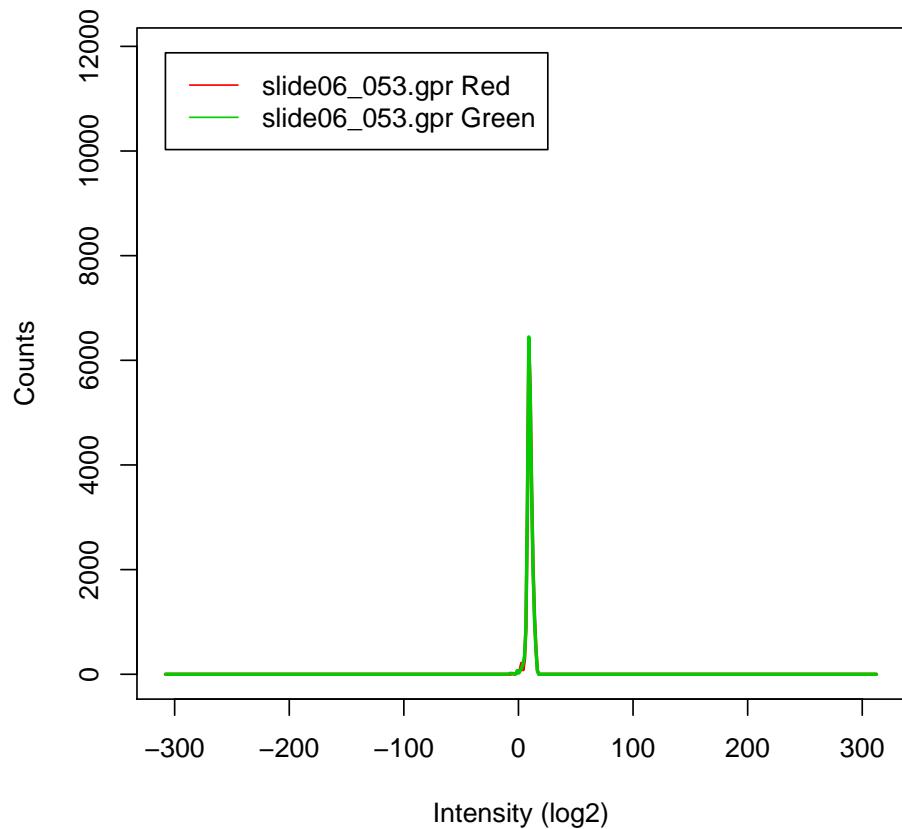


Figure 1.160: Histogram of the array 6 (slide06_053.gpr). Between array normalized data.

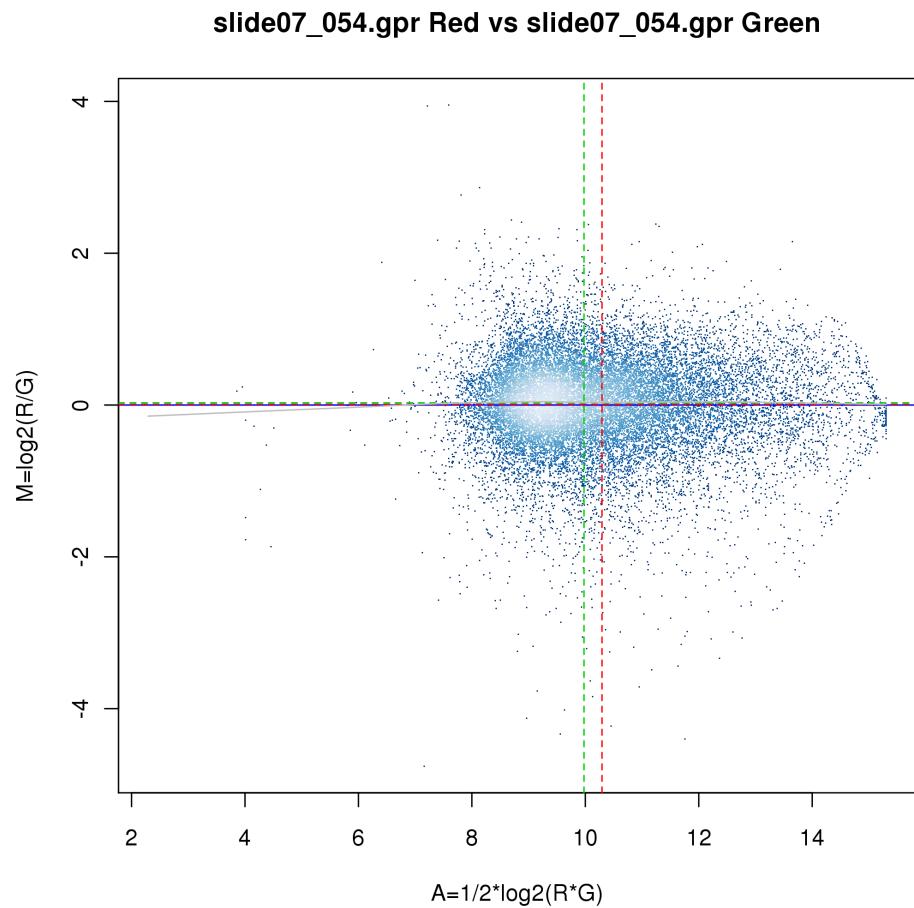


Figure 1.161: MA plot of array 7 (slide07_054.gpr). Between array normalized data.

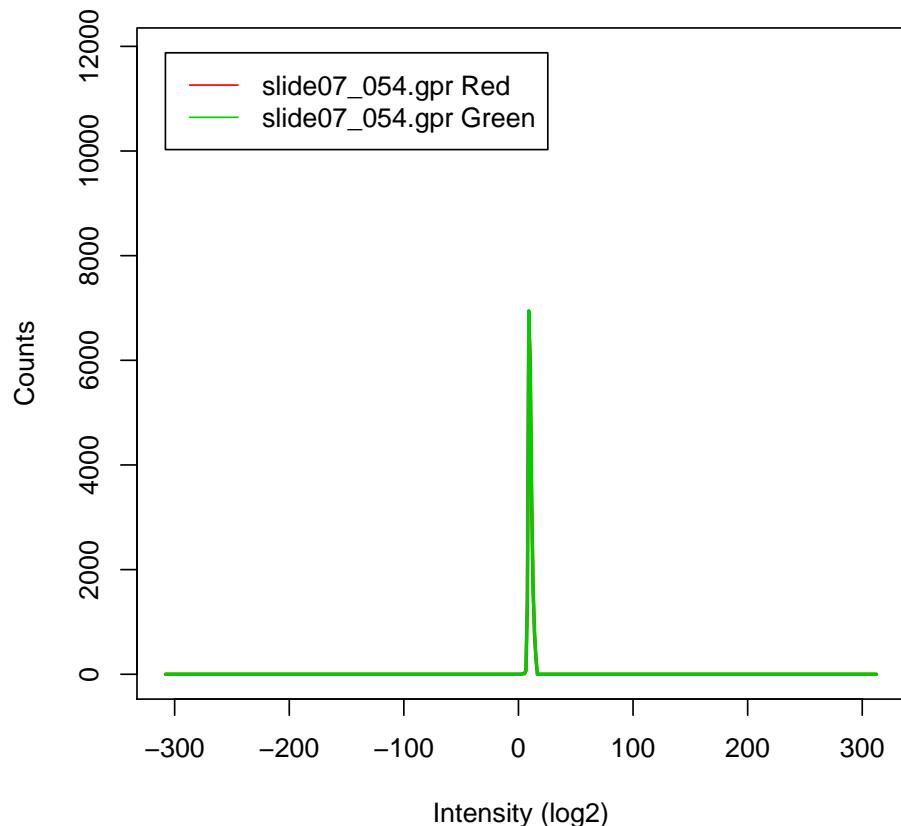


Figure 1.162: Histogram of the array 7 (slide07_054.gpr). Between array normalized data.

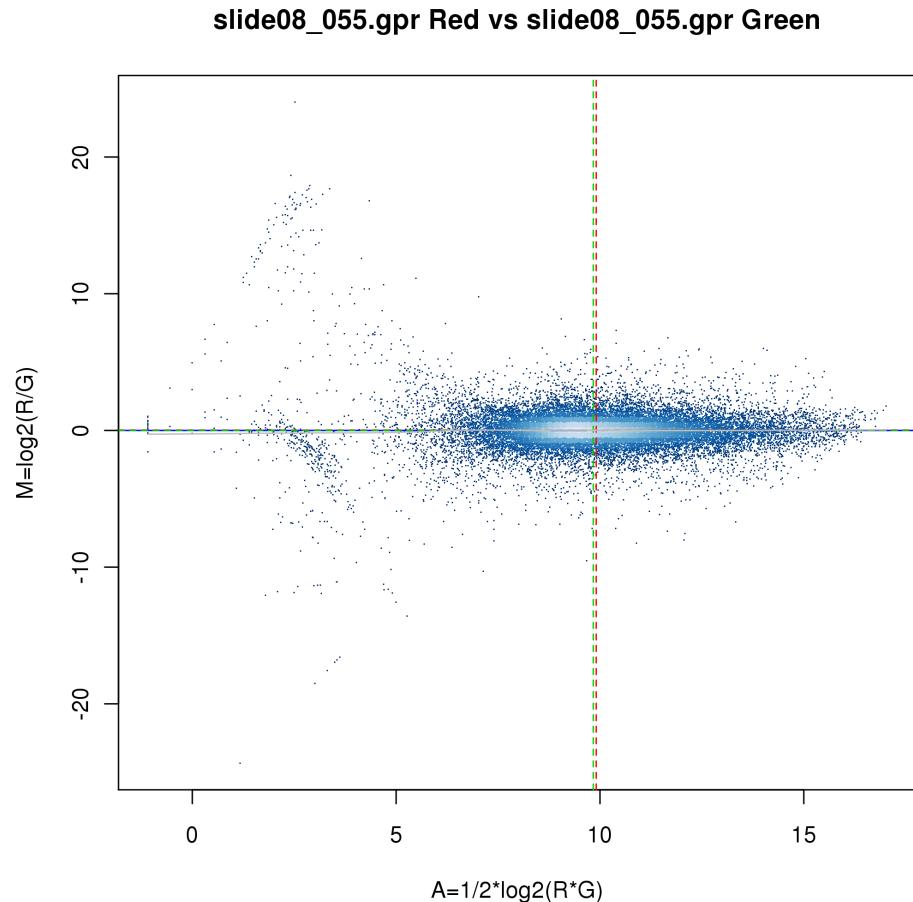


Figure 1.163: MA plot of array 8 (slide08_055.gpr). Between array normalized data.

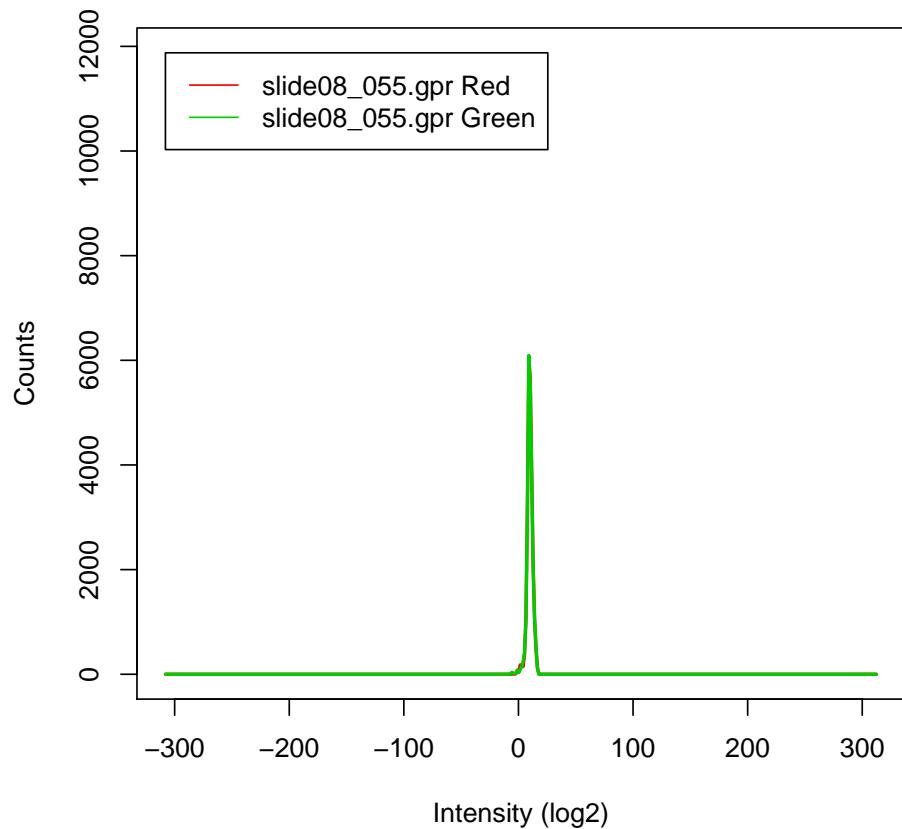


Figure 1.164: Histogram of the array 8 (slide08_055.gpr). Between array normalized data.

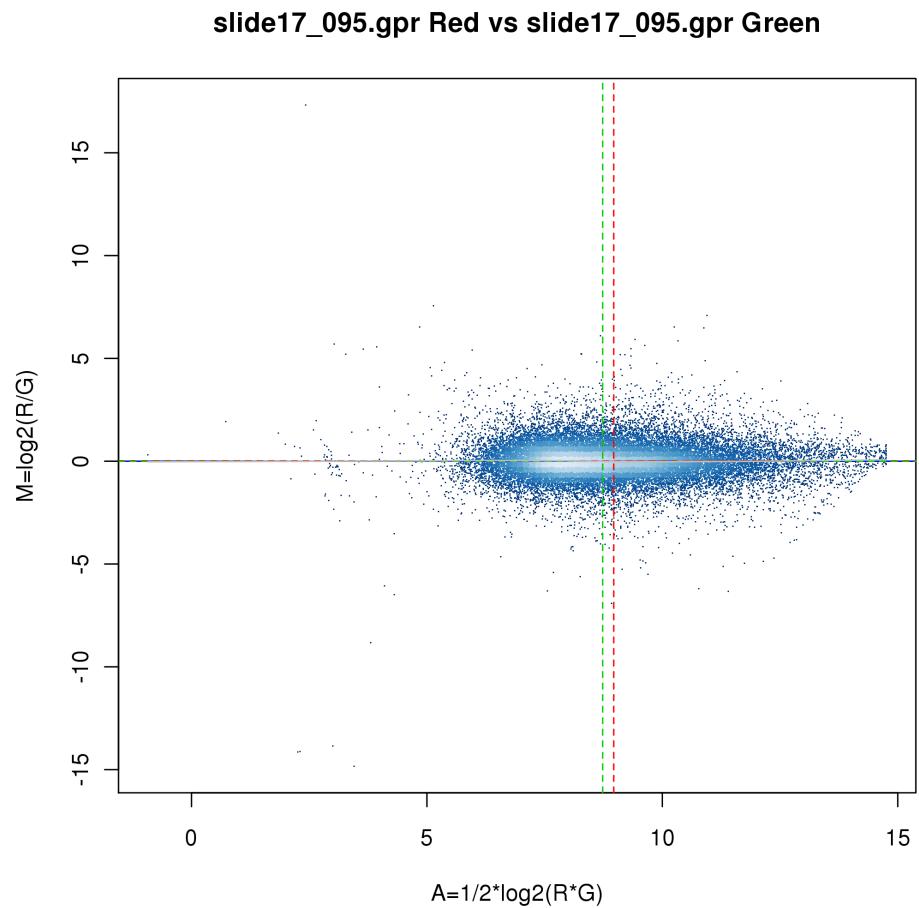


Figure 1.165: MA plot of array 9 (slide17_095.gpr). Between array normalized data.

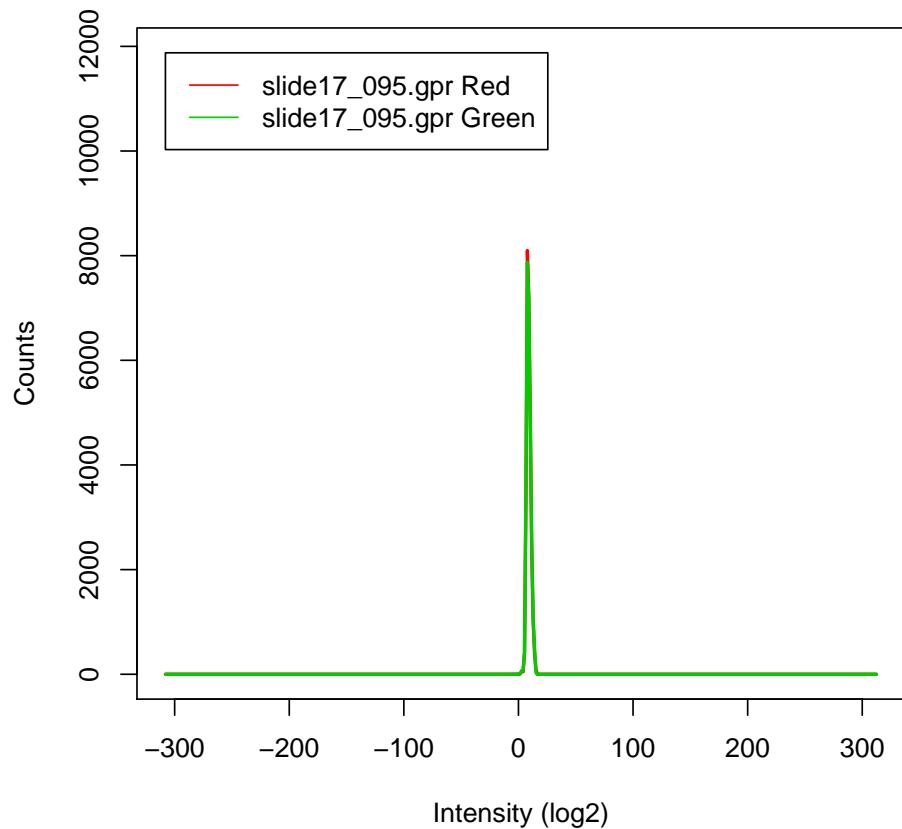


Figure 1.166: Histogram of the array 9 (slide17_095.gpr). Between array normalized data.

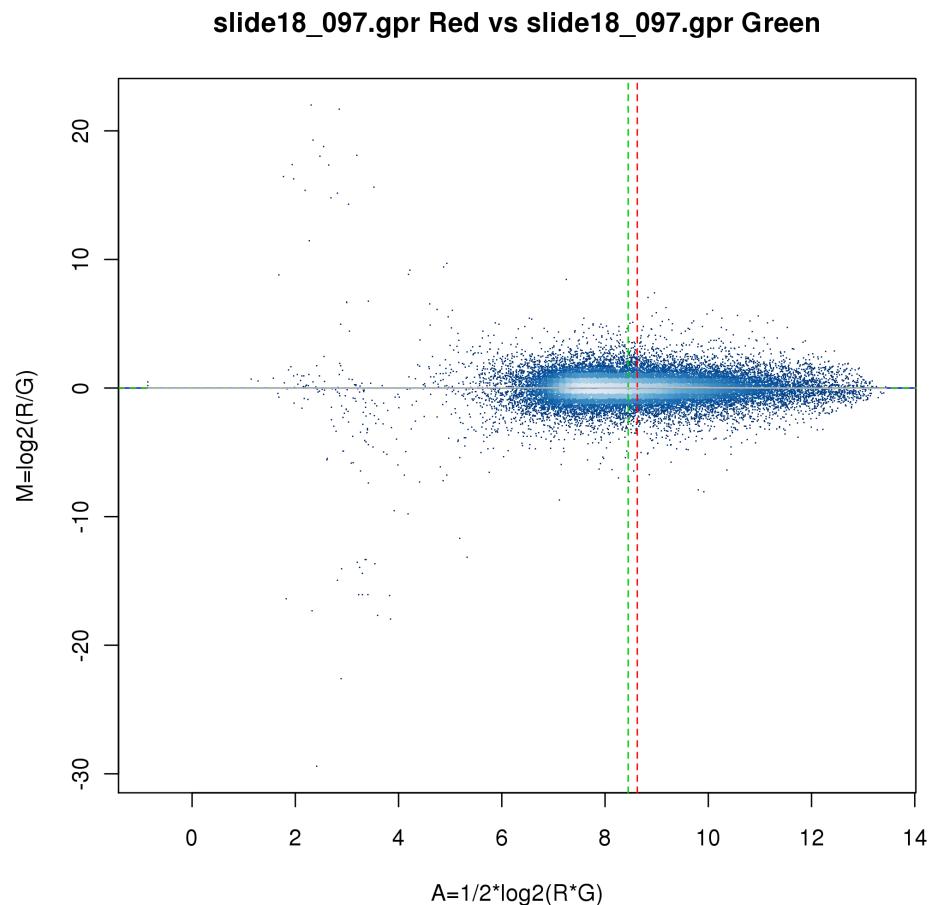


Figure 1.167: MA plot of array 10 (slide18_097.gpr). Between array normalized data.

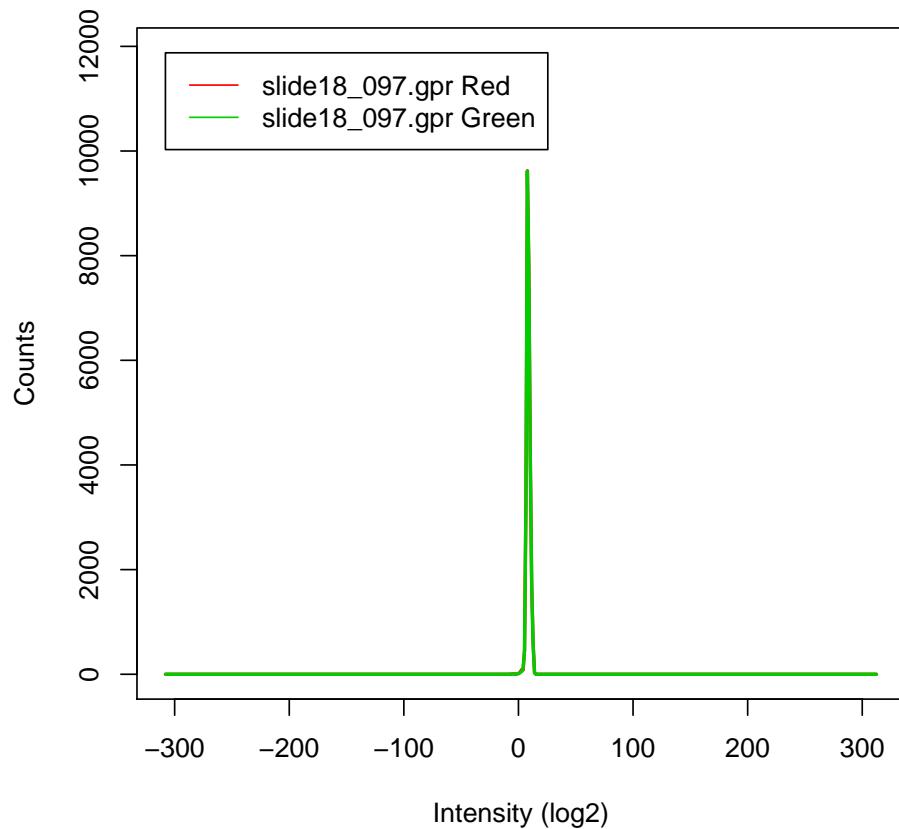


Figure 1.168: Histogram of the array 10 (slide18_097.gpr). Between array normalized data.

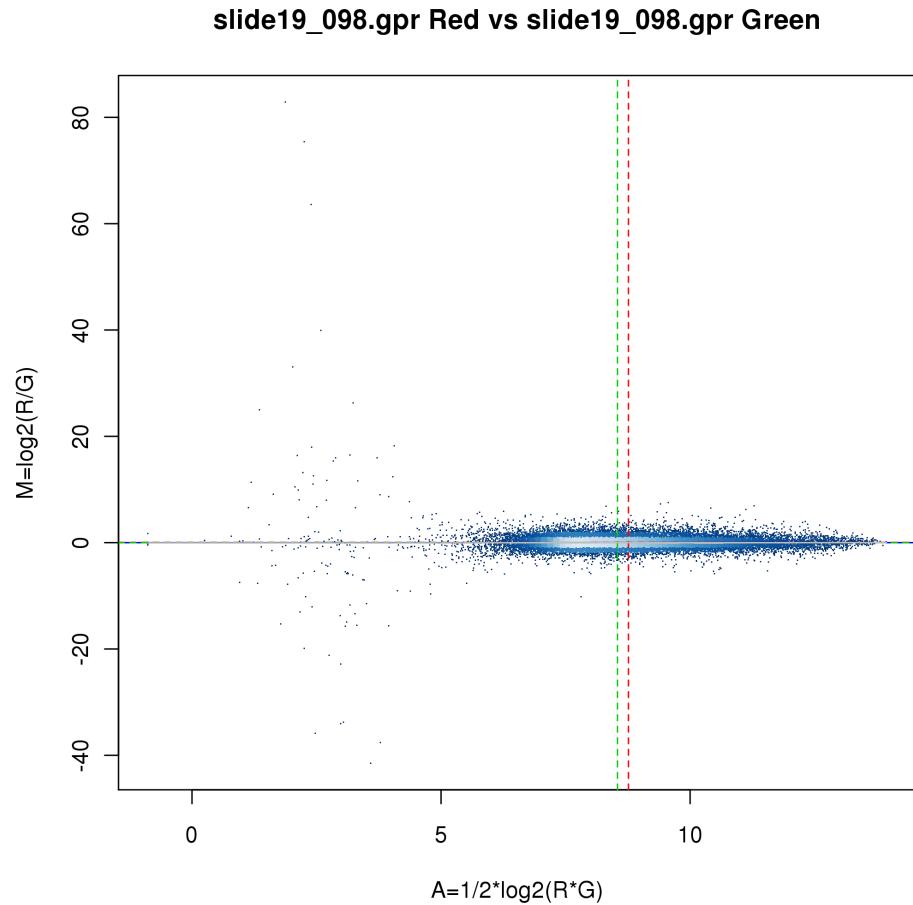


Figure 1.169: MA plot of array 11 (slide19_098.gpr). Between array normalized data.

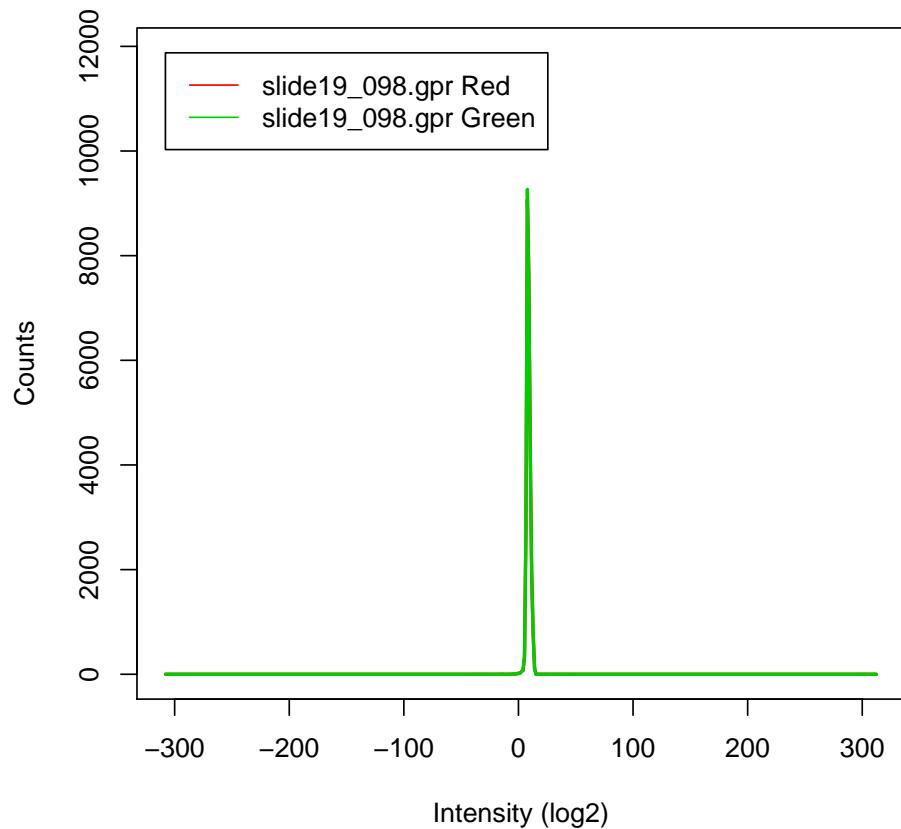


Figure 1.170: Histogram of the array 11 (slide19_098.gpr). Between array normalized data.

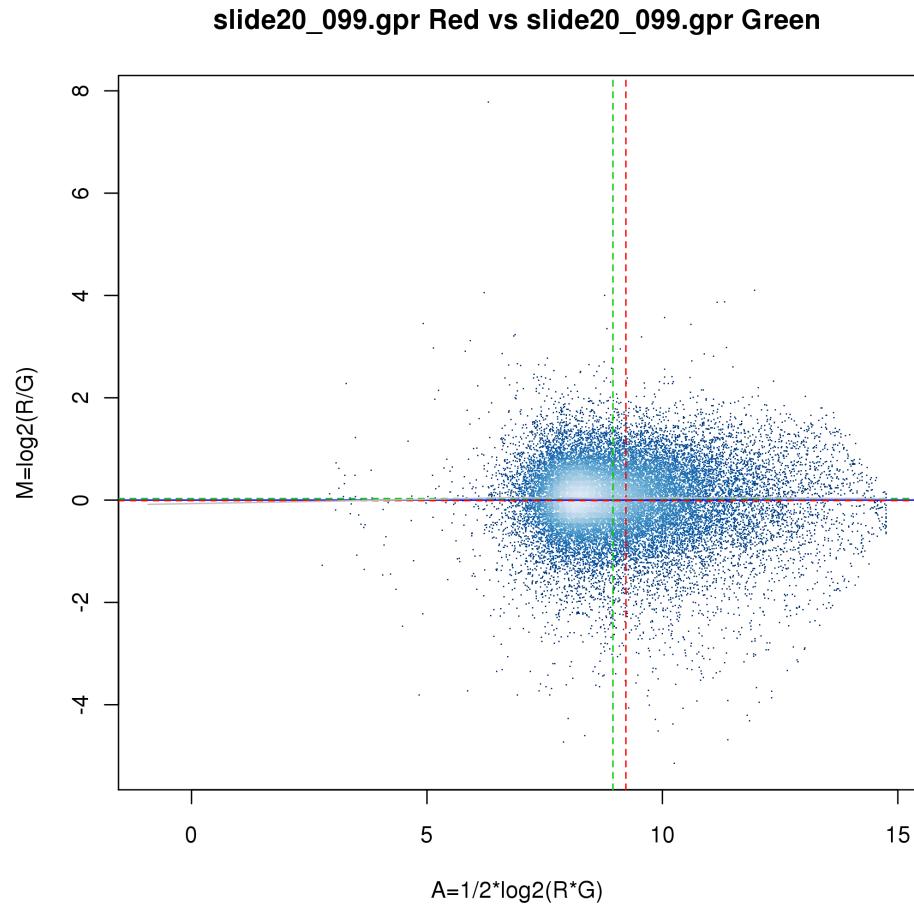


Figure 1.171: MA plot of array 12 (slide20_099.gpr). Between array normalized data.

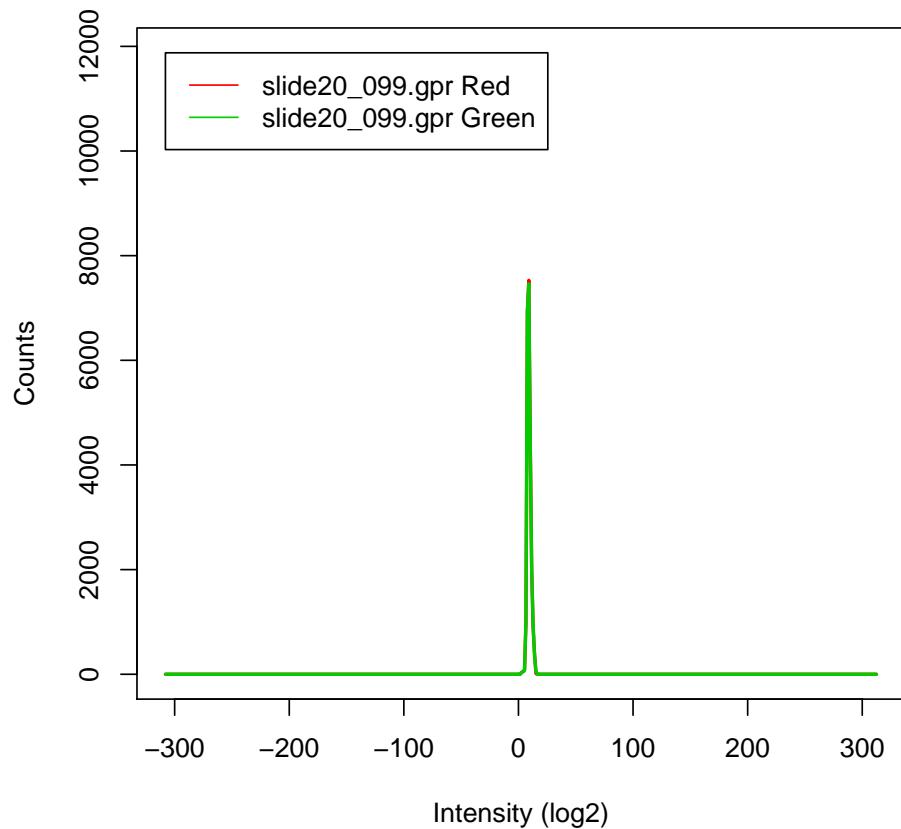


Figure 1.172: Histogram of the array 12 (slide20_099.gpr). Between array normalized data.

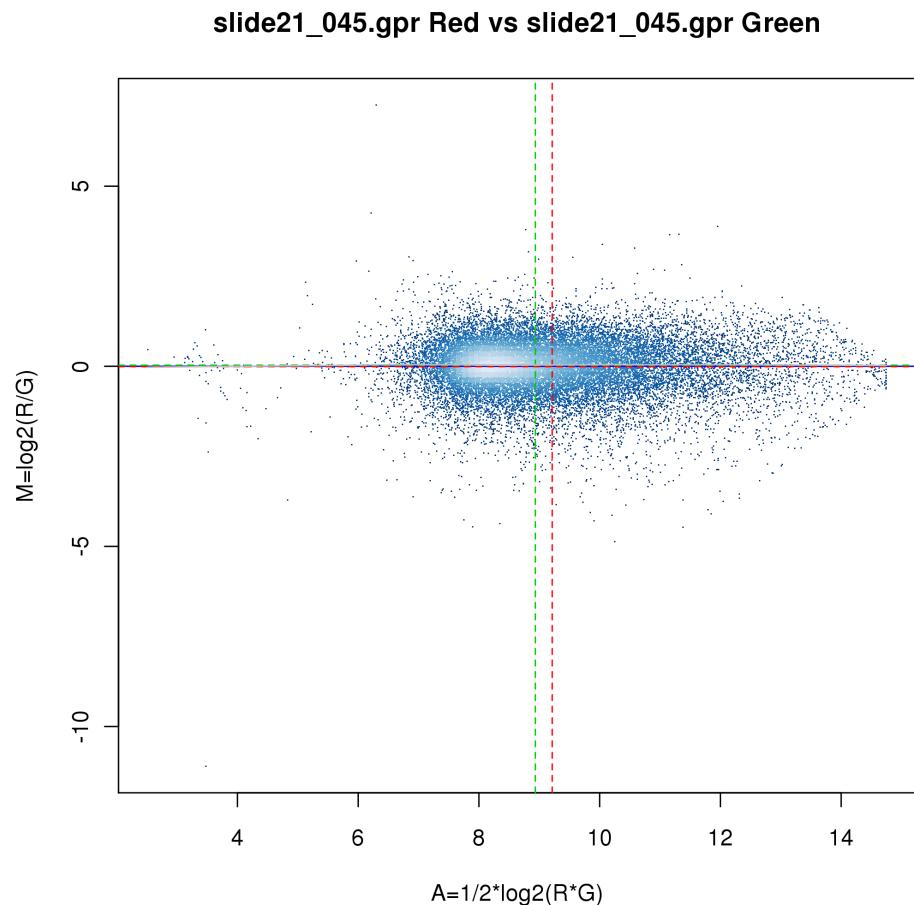


Figure 1.173: MA plot of array 13 (slide21_045.gpr). Between array normalized data.

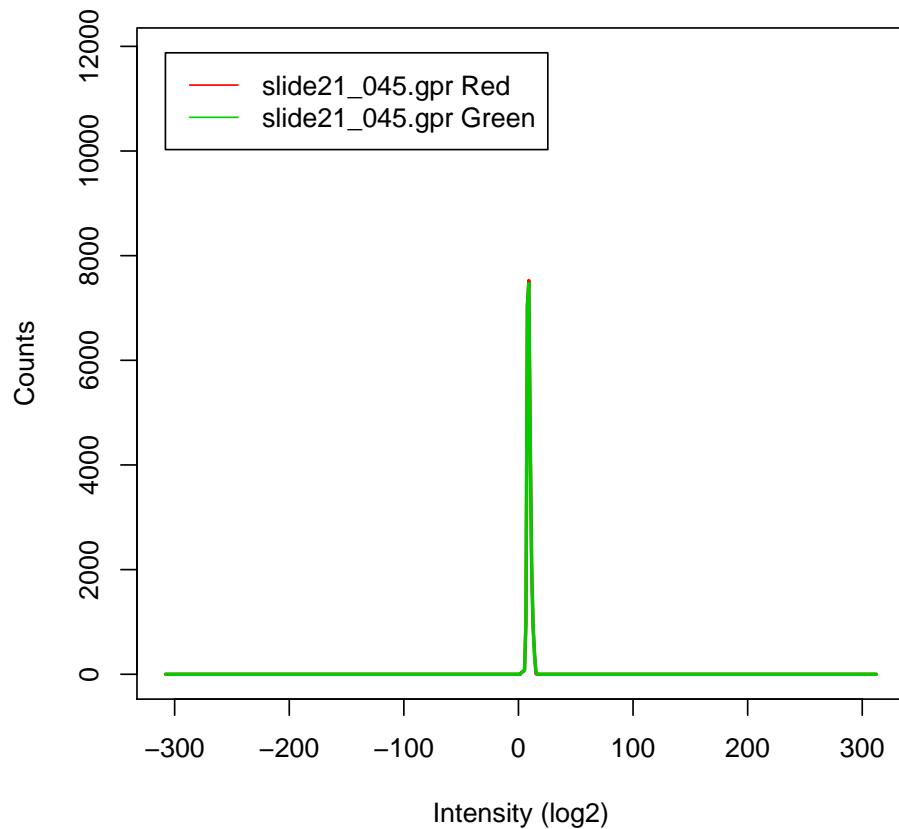


Figure 1.174: Histogram of the array 13 (slide21_045.gpr). Between array normalized data.

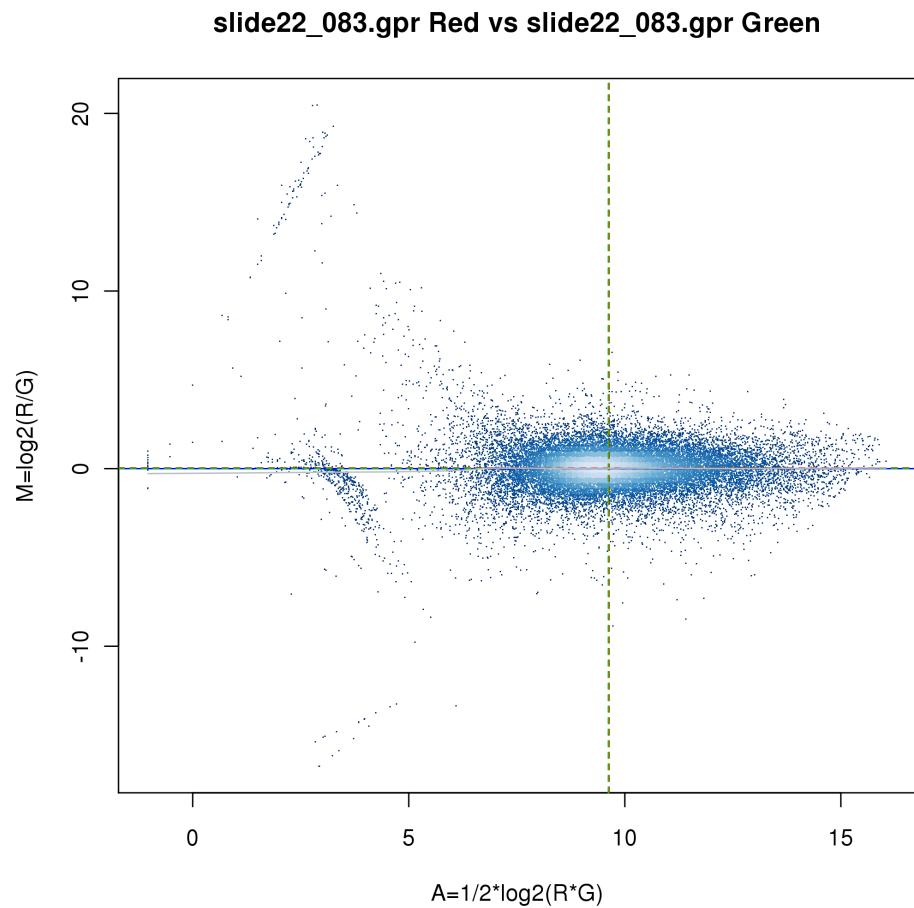


Figure 1.175: MA plot of array 14 (slide22_083.gpr). Between array normalized data.

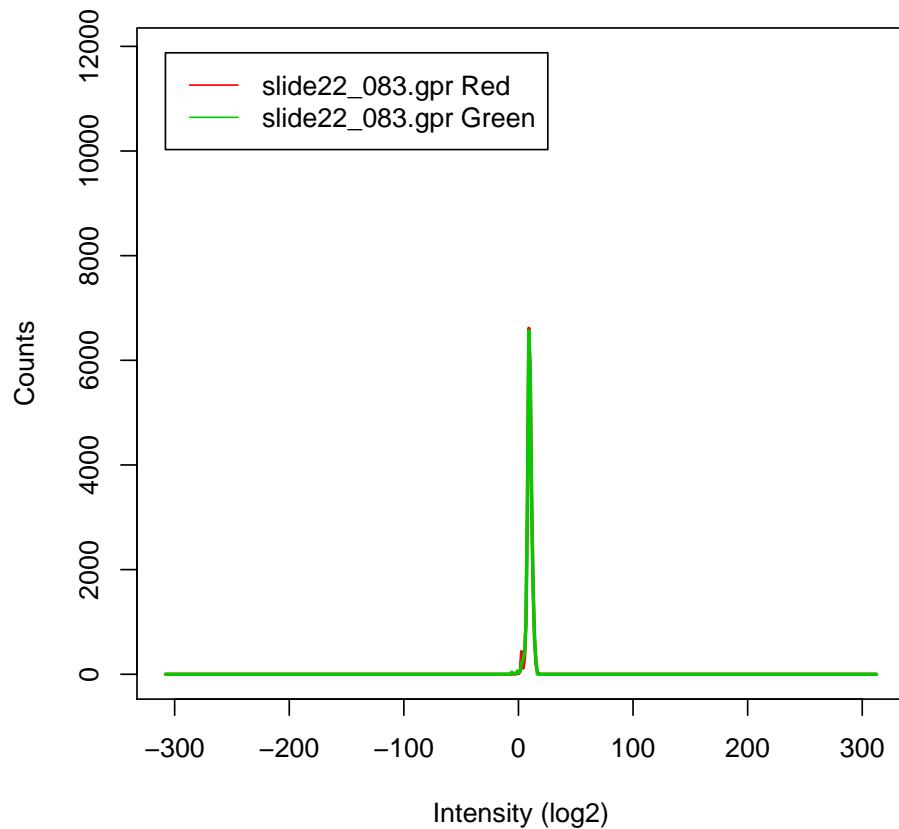


Figure 1.176: Histogram of the array 14 (slide22_083.gpr). Between array normalized data.

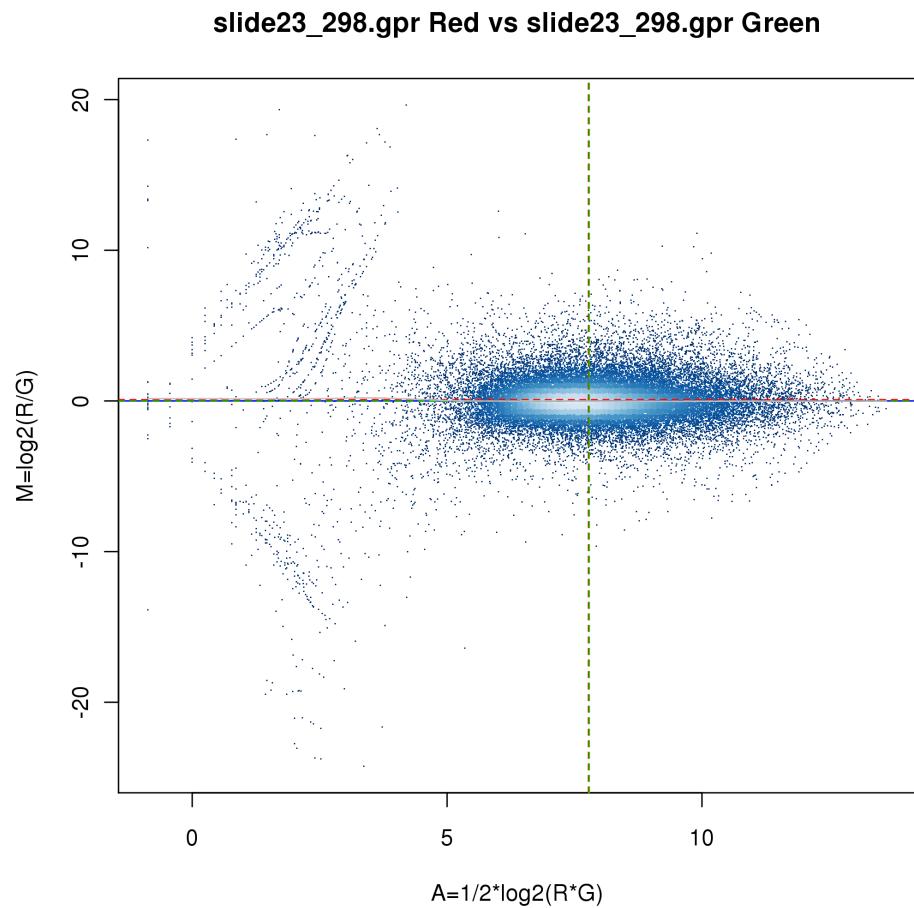


Figure 1.177: MA plot of array 15 (slide23_298.gpr). Between array normalized data.

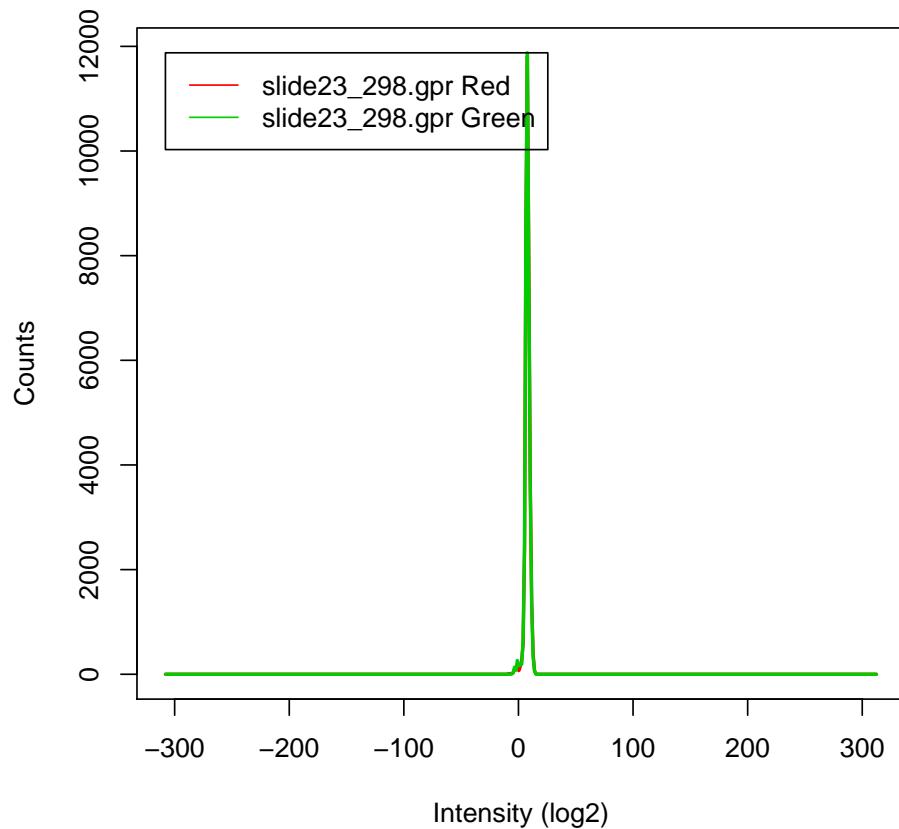


Figure 1.178: Histogram of the array 15 (slide23_298.gpr). Between array normalized data.

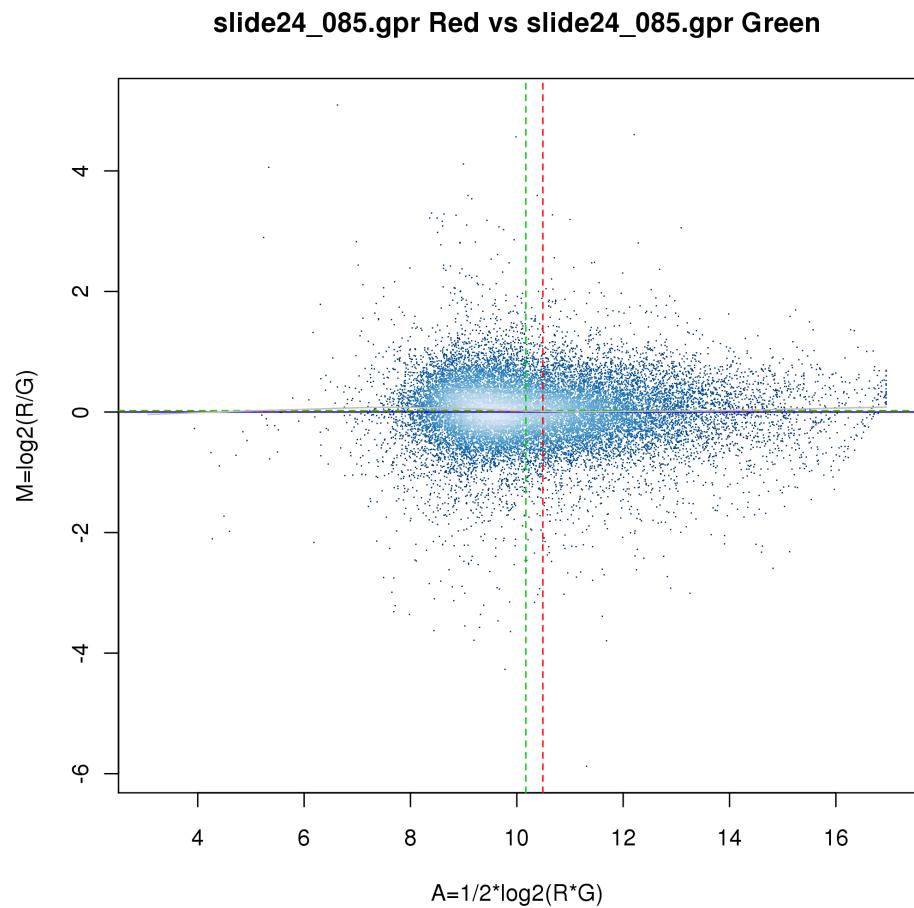


Figure 1.179: MA plot of array 16 (slide24_085.gpr). Between array normalized data.

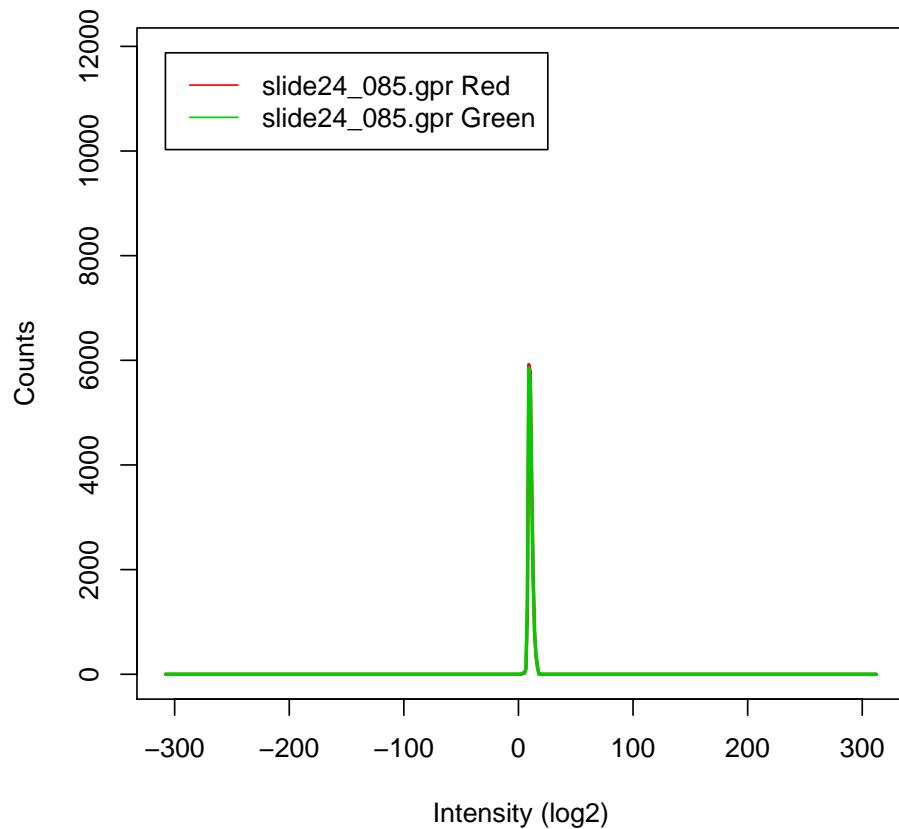


Figure 1.180: Histogram of the array 16 (slide24_085.gpr). Between array normalized data.

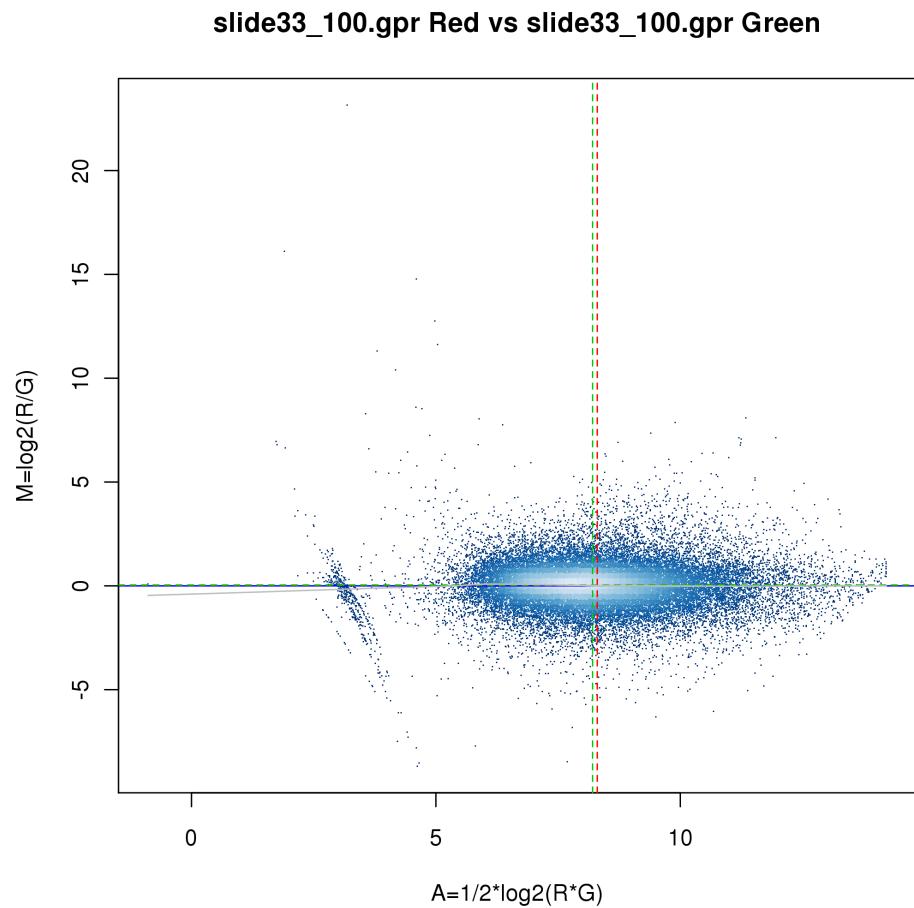


Figure 1.181: MA plot of array 17 (slide33_100.gpr). Between array normalized data.

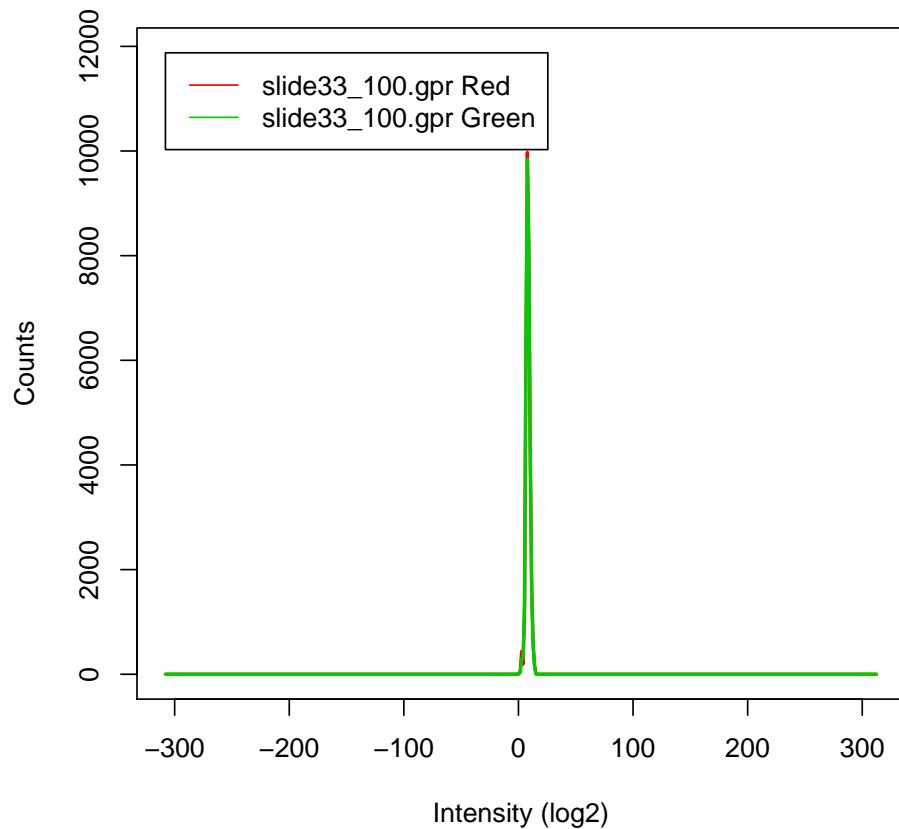


Figure 1.182: Histogram of the array 17 (slide33_100.gpr). Between array normalized data.

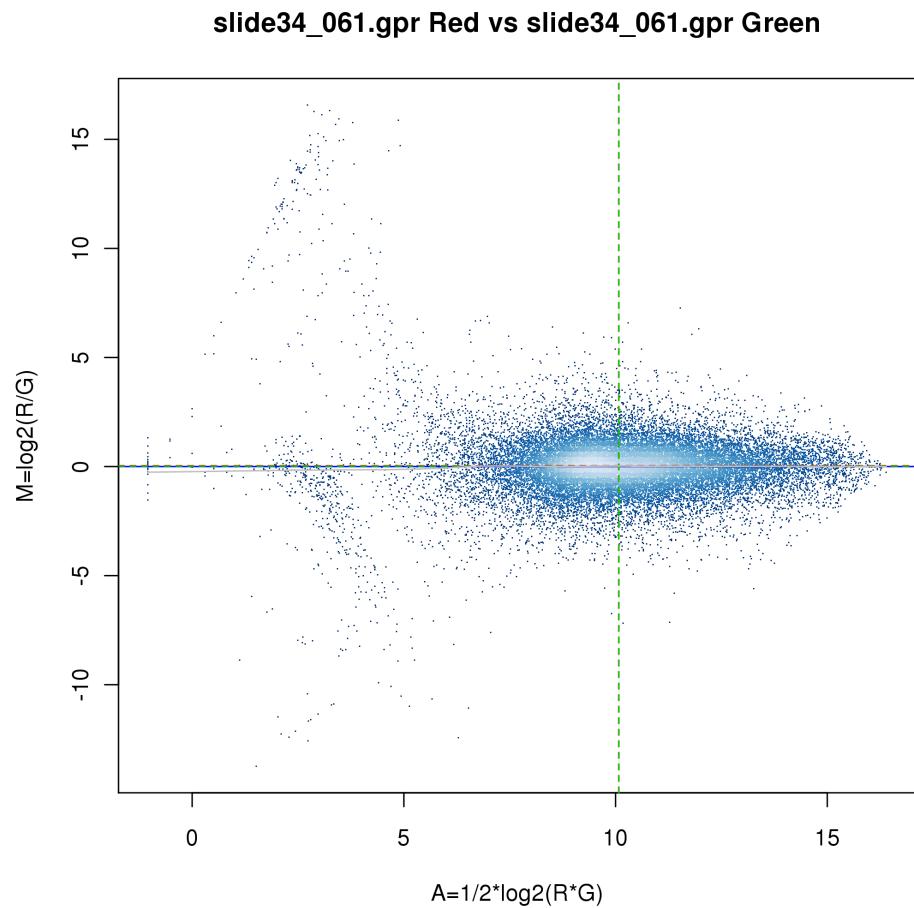


Figure 1.183: MA plot of array 18 (slide34_061.gpr). Between array normalized data.

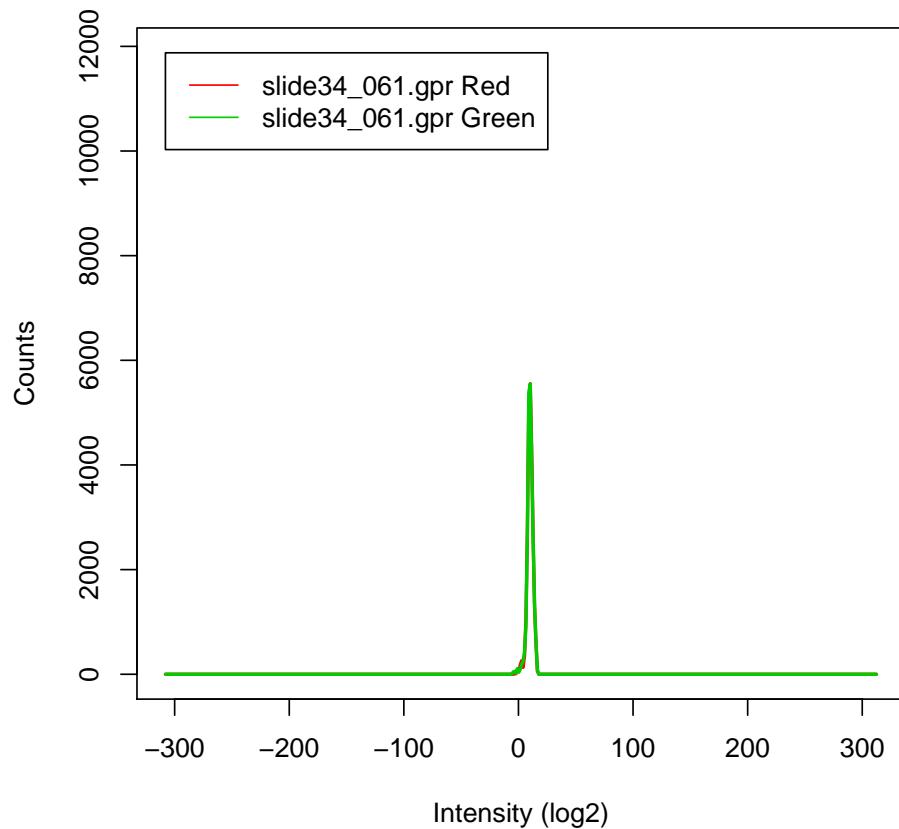


Figure 1.184: Histogram of the array 18 (slide34_061.gpr). Between array normalized data.

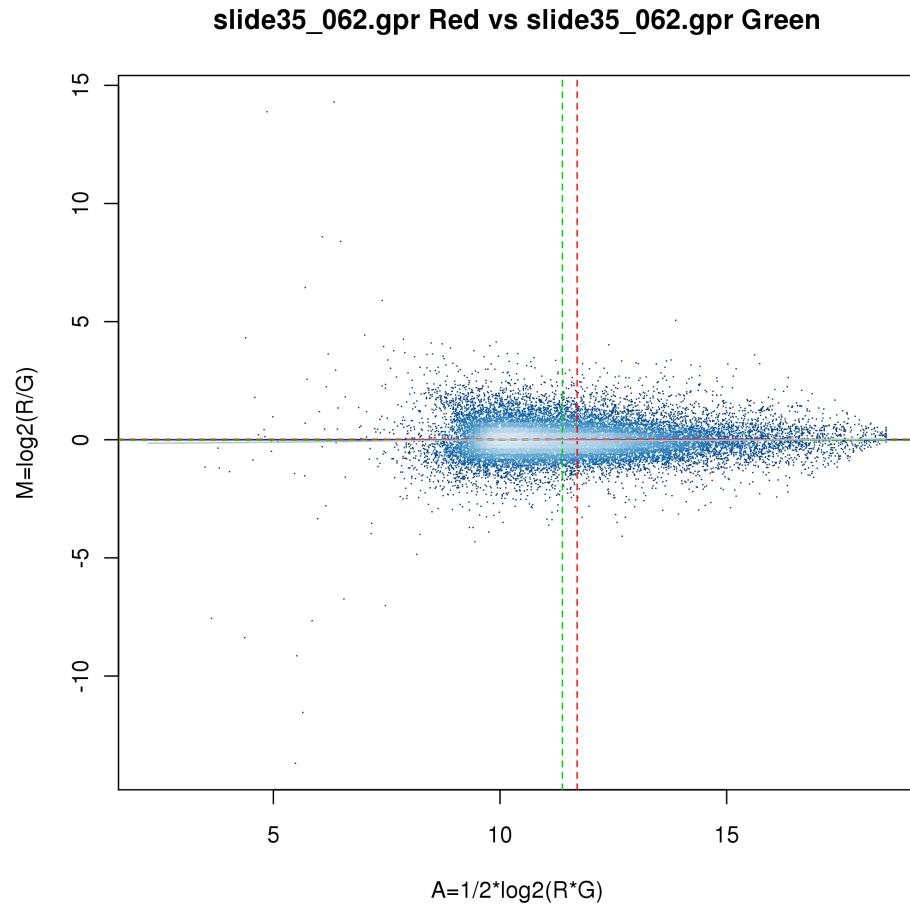


Figure 1.185: MA plot of array 19 (slide35_062.gpr). Between array normalized data.

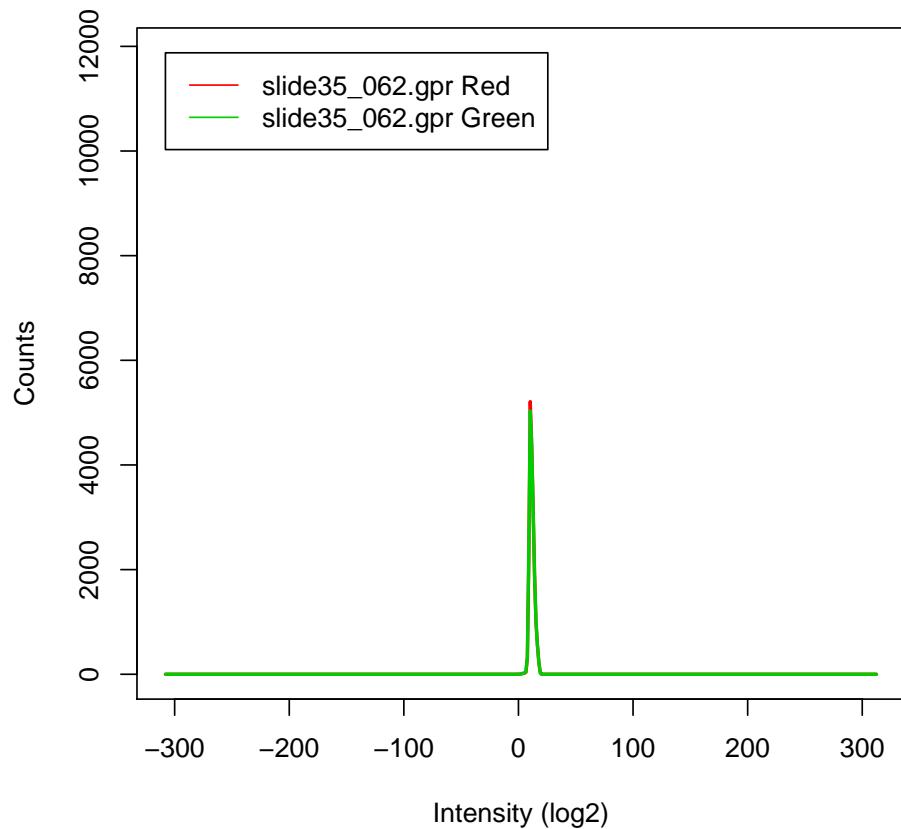


Figure 1.186: Histogram of the array 19 (slide35_062.gpr). Between array normalized data.

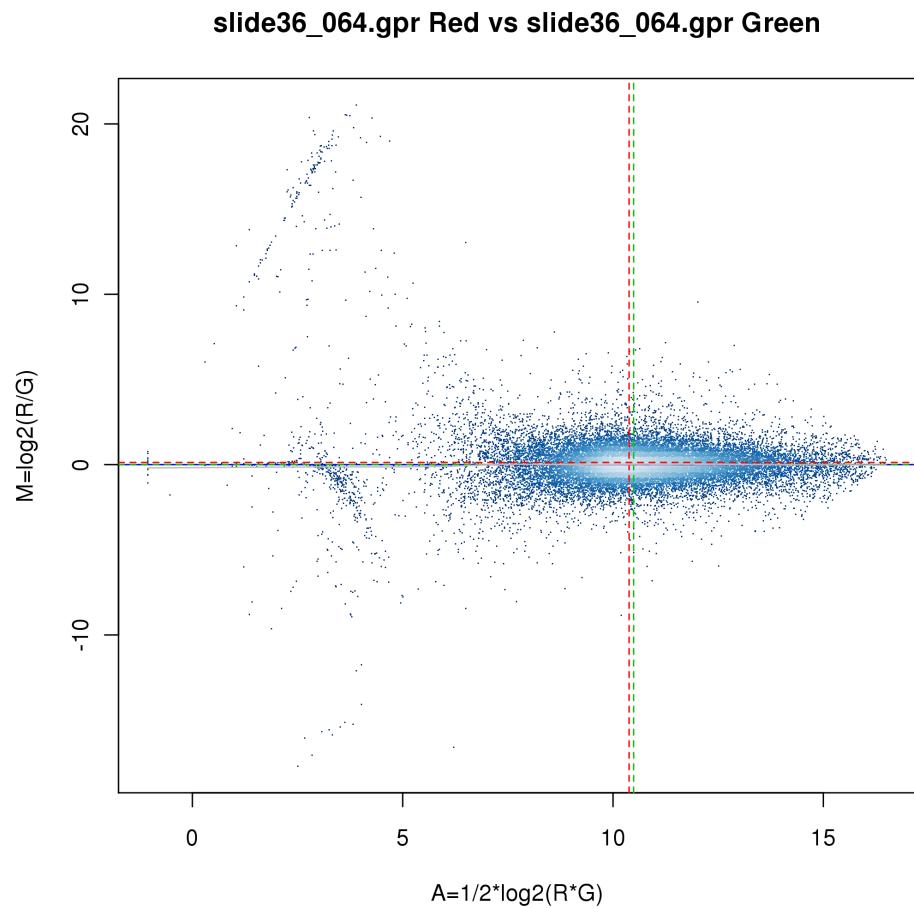


Figure 1.187: MA plot of array 20 (slide36_064.gpr). Between array normalized data.

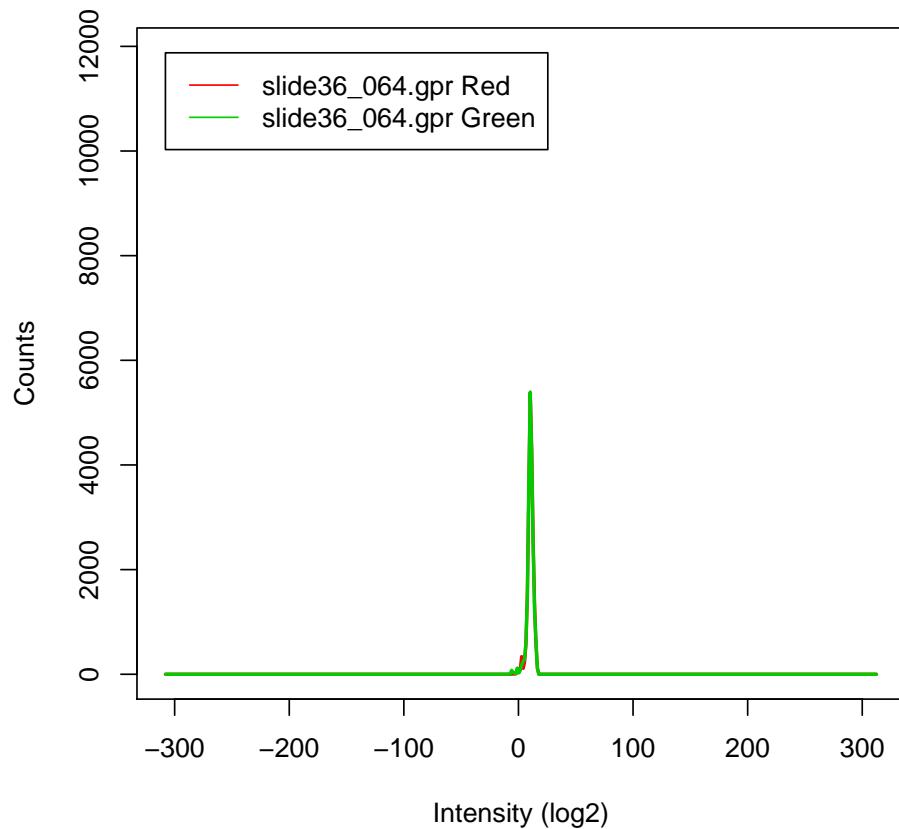


Figure 1.188: Histogram of the array 20 (slide36_064.gpr). Between array normalized data.

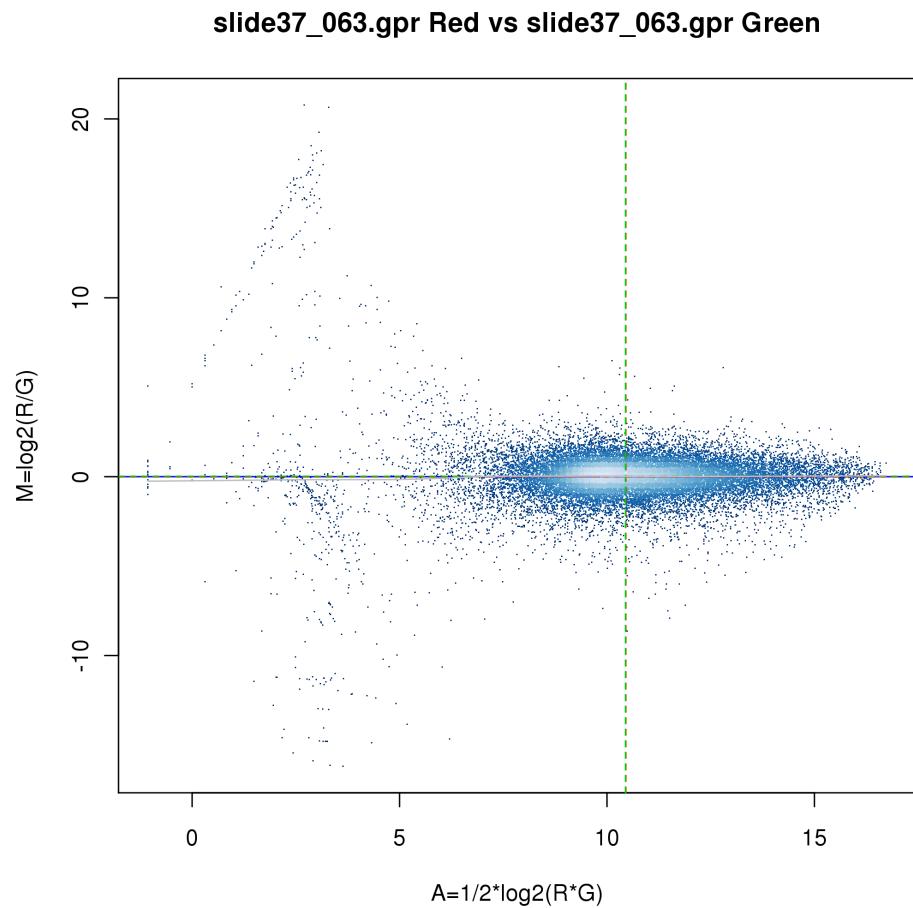


Figure 1.189: MA plot of array 21 (slide37_063.gpr). Between array normalized data.

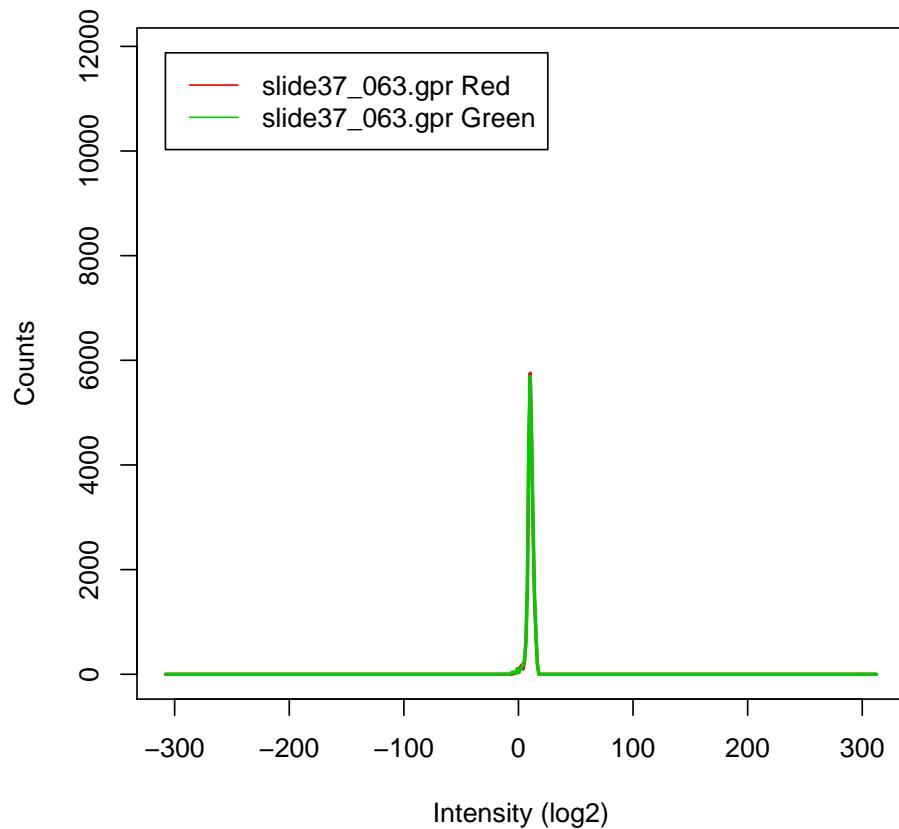


Figure 1.190: Histogram of the array 21 (slide37_063.gpr). Between array normalized data.

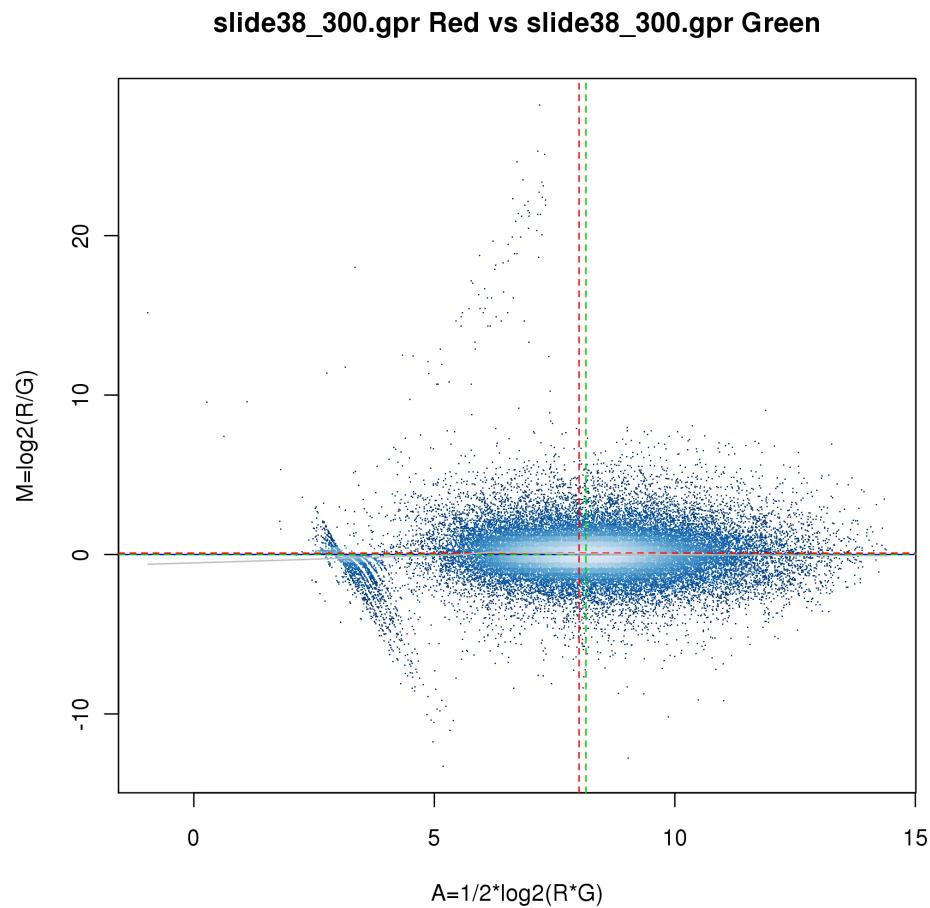


Figure 1.191: MA plot of array 22 (slide38_300.gpr). Between array normalized data.

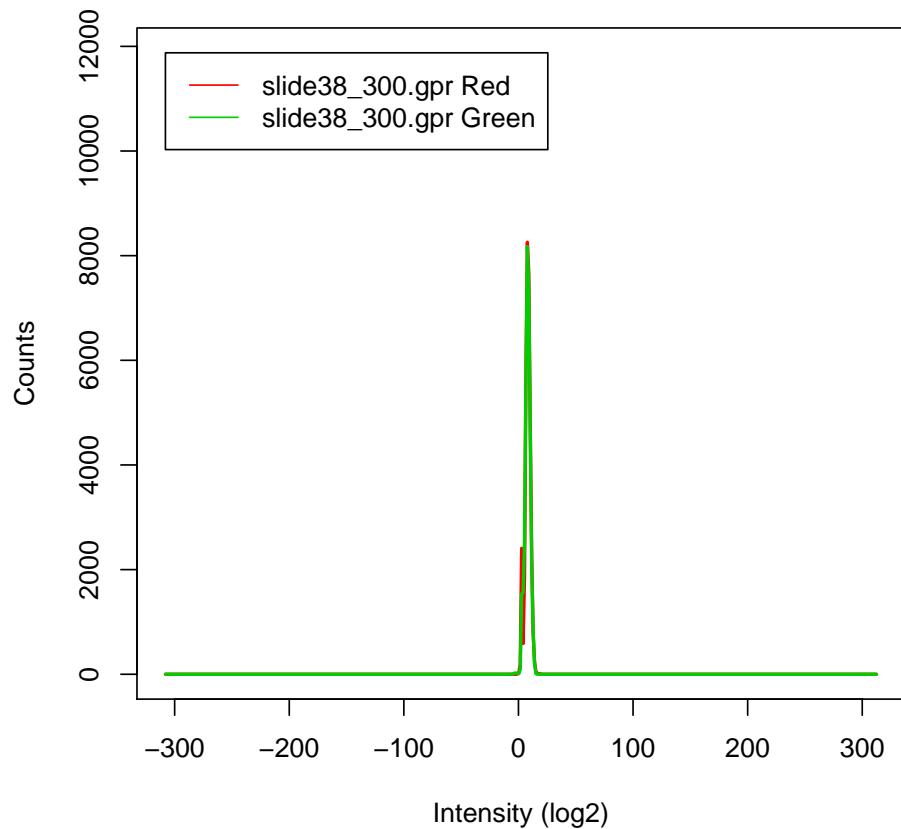


Figure 1.192: Histogram of the array 22 (slide38_300.gpr). Between array normalized data.

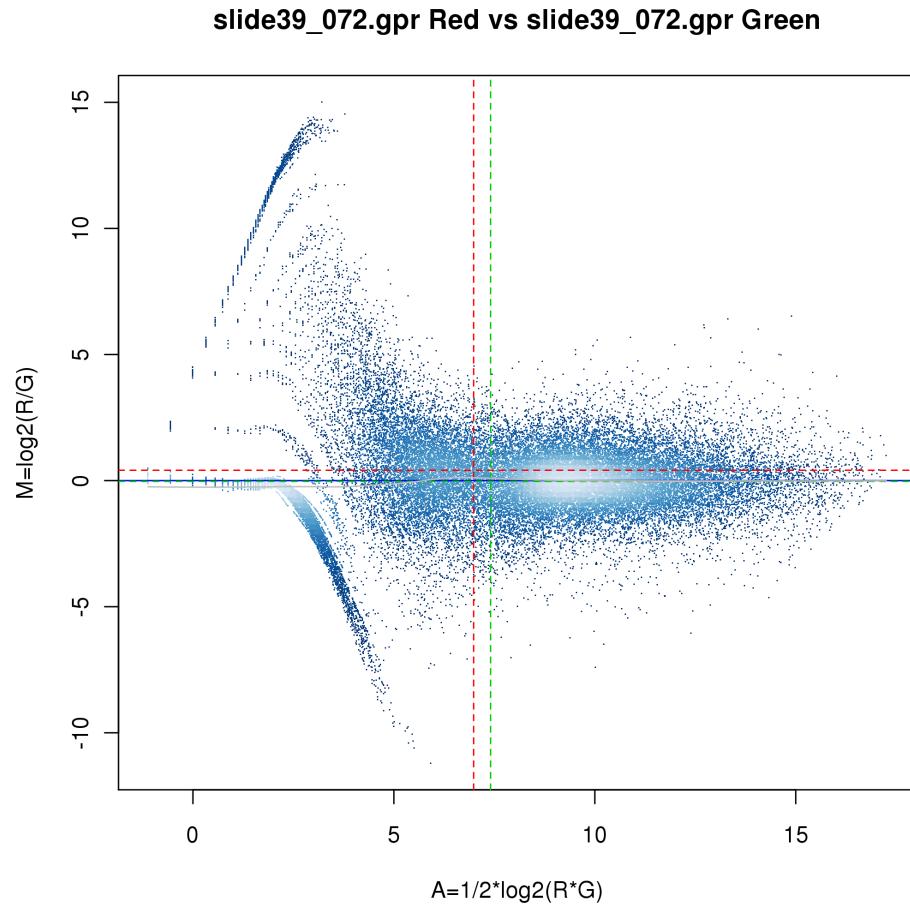


Figure 1.193: MA plot of array 23 (slide39_072.gpr). Between array normalized data.

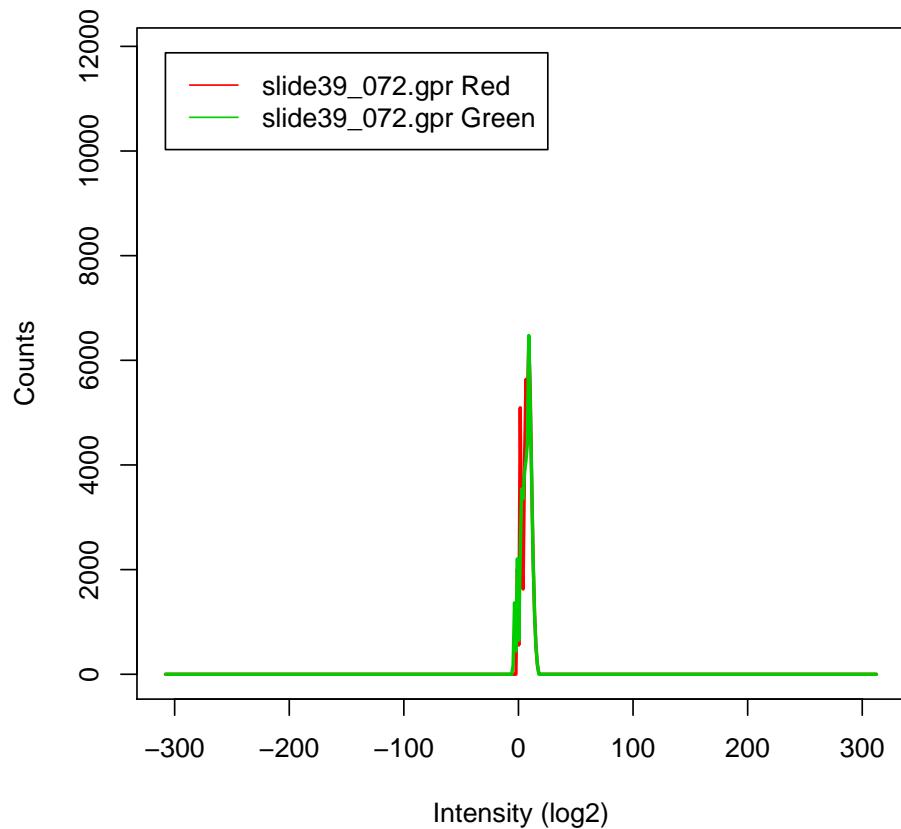


Figure 1.194: Histogram of the array 23 (slide39_072.gpr). Between array normalized data.

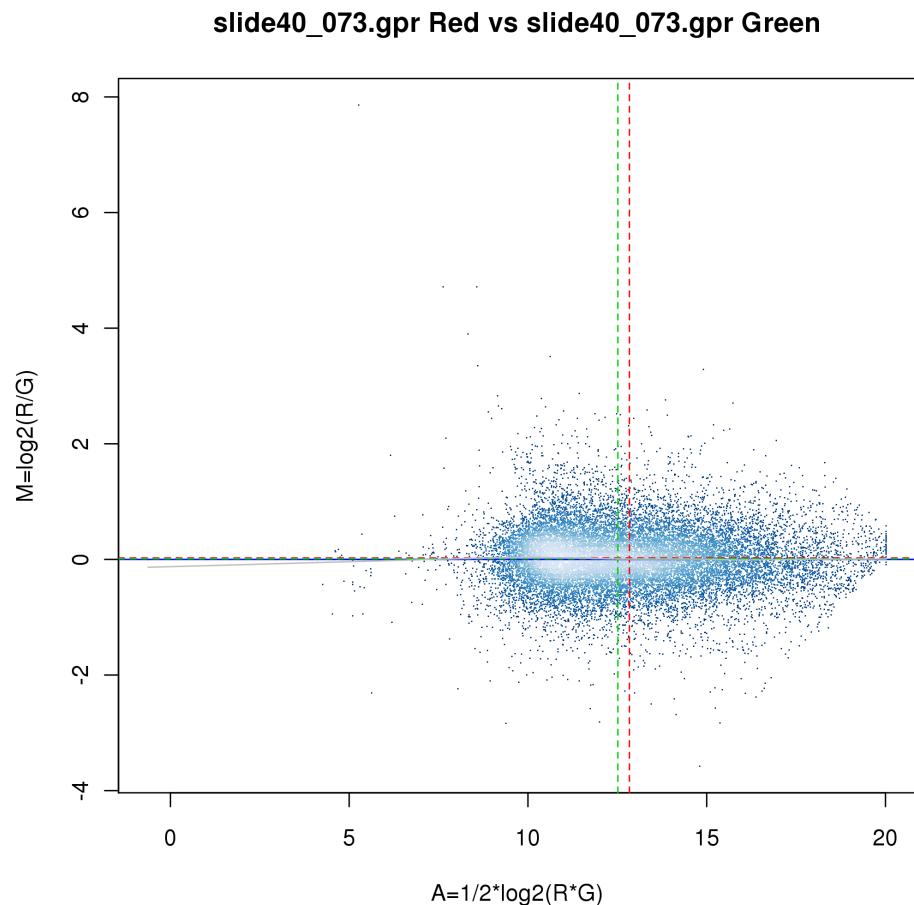


Figure 1.195: MA plot of array 24 (slide40_073.gpr). Between array normalized data.

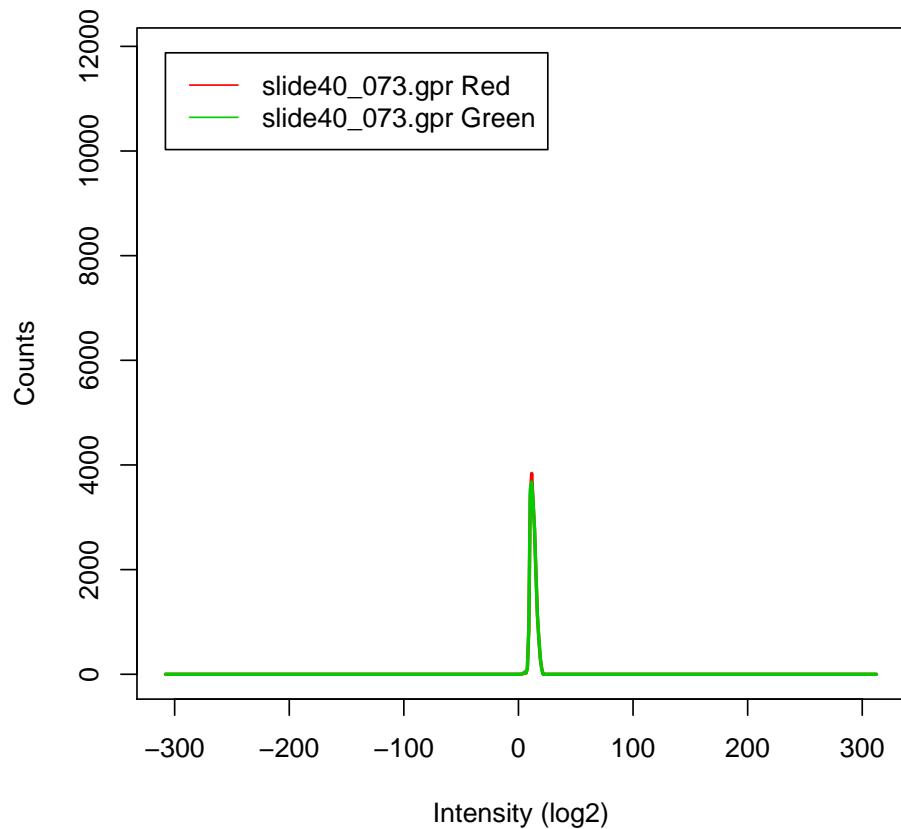


Figure 1.196: Histogram of the array 24 (slide40_073.gpr). Between array normalized data.

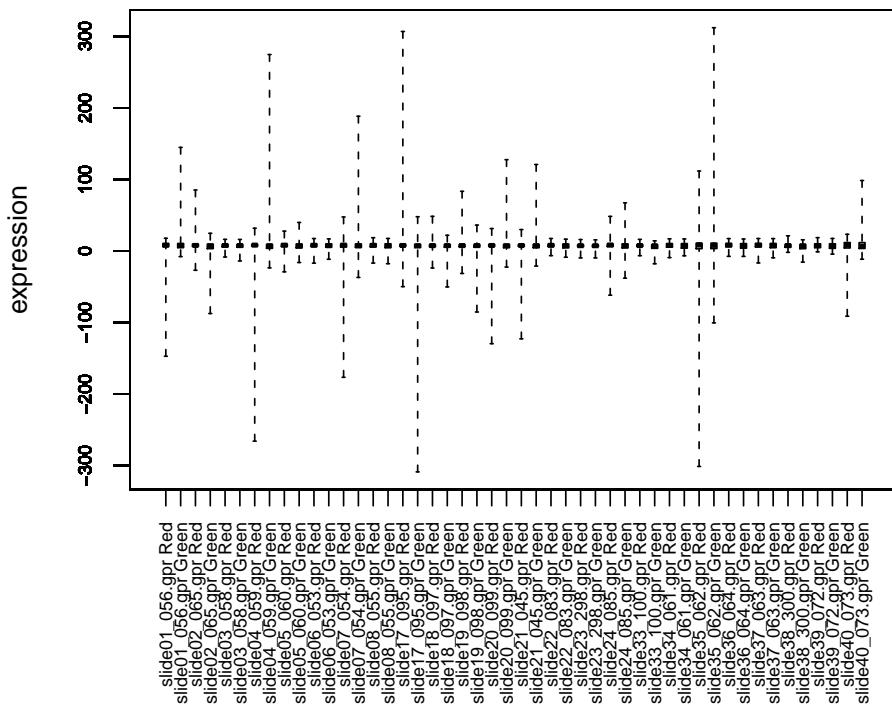


Figure 1.197: Boxplots of the signal intensities of each signal channel of the microarrays. Between array normalized data.

```
> drawHistogram(Dummy, lwd = 2, col = rep(c(2, 3), 24))  
calculating histograms
```

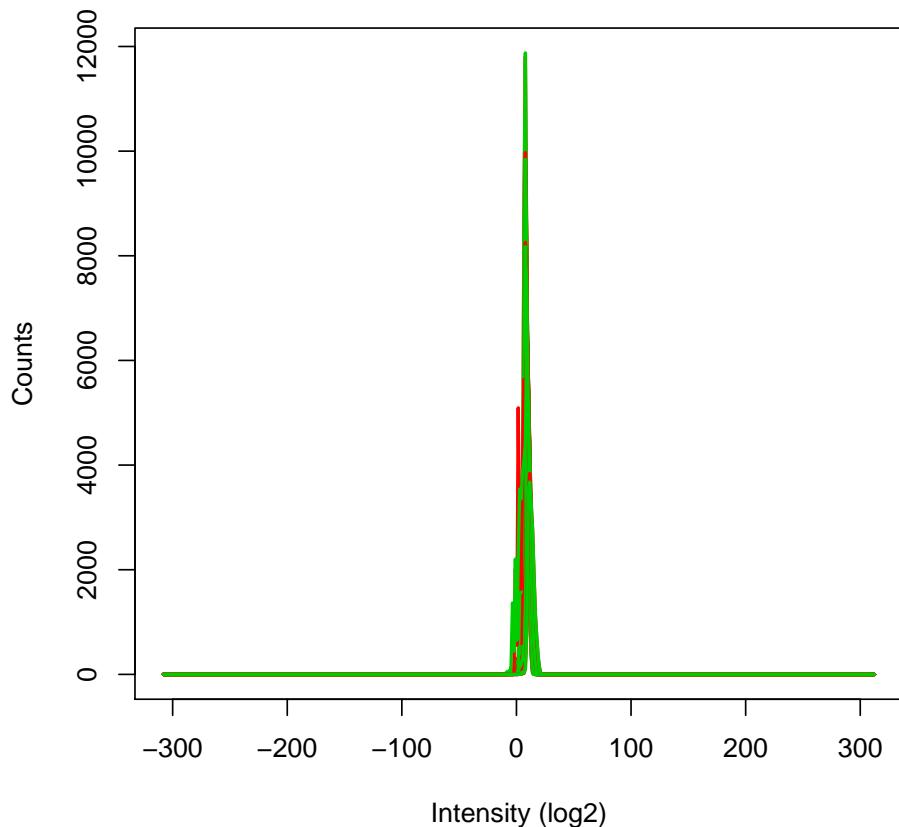


Figure 1.198: Histogram of all arrays within this experiment after the between-array-normalization. The green lines corresponds to the green signal channels and the red line to the red channel respectively.

1.5 Saving the normalized M and A values

The normalized M and A values of the arrays within this experiment are saved to a tab delimited txt file. The M and A values were saved to the file: *NORMALIZATION.txt*

Chapter 2

Replicate handling

In the replication handling step the normalized intensities of replicated spots within an array and between technical replicated (or dye swapped) arrays are averaged to one single value per gene. The *merged arrays* were defined as follows:

- *AB* contains the averaged values of the replicates: slide01_056.gpr, slide17_095.gpr, slide33_100.gpr,
- *BC* contains the averaged values of the replicates: slide02_065.gpr, slide18_097.gpr, slide34_061.gpr,
- *CD* contains the averaged values of the replicates: slide03_058.gpr, slide19_098.gpr, slide35_062.gpr,
- *DE* contains the averaged values of the replicates: slide04_059.gpr, slide20_099.gpr, slide36_064.gpr,
- *EF* contains the averaged values of the replicates: slide05_060.gpr, slide21_045.gpr, slide37_063.gpr,
- *FG* contains the averaged values of the replicates: slide06_053.gpr, slide22_083.gpr, slide38_300.gpr,
- *GH* contains the averaged values of the replicates: slide07_054.gpr, slide23_298.gpr, slide39_072.gpr,
- *HA* contains the averaged values of the replicates: slide08_055.gpr, slide24_085.gpr, slide40_073.gpr,

```
> source("utils.R")
> Eset <- newMadbSet(Slides.norm)

Converting a limma MAList into a MadbSet...
Setting the weights... a weights of 0 means the gene was flagged, a weights of one means the signal is ok!
```

```
Inserting available annotation into the slot @genes

Inserting available annotation into the slot @genes

> rm(Slides.norm)
> g <- gc()
```

Before replicate handling, your experiment consisted of 24 arrays with each 46128 spots.

```
> Eset <- average(Eset, average.which = c(1, 2, 3, 4, 5, 6, 7,
+   8, 9, 10, 11, 12, 13, 14, 15, 16, 1, 2, 3, 4, 5, 6, 7, 8,
+   9, 10, 11, 12, 13, 14, 15, 16, 1, 2, 3, 4, 5, 6, 7, 8, 9,
+   10, 11, 12, 13, 14, 15, 16), method = "mean", array.names = c("AB Red",
+   "AB Green", "BC Red", "BC Green", "CD Red", "CD Green", "DE Red",
+   "DE Green", "EF Red", "EF Green", "FG Red", "FG Green", "GH Red",
+   "GH Green", "HA Red", "HA Green"), average.genes = TRUE,
+   exclude.flagged = TRUE, only.good.spots = TRUE)

48 columns where combined to 16 columns
averaging replicated genes per array
46128 rows where combined to 43441 rows
```

After combining replicated arrays and spots your experiment consists of 8 arrays with each 43441 spots (corresponding to unique sequences).

Chapter 3

Determining differentially expressed genes using test statistics

Statistical tests are used in this chapter to define genes that are differentially expressed between two groups in this micro array experiment. Statistical tests allow to find genes, that show different expression levels between two sample groups and small alterations in expression levels within each group. The statistical tests used in this analysis are mainly provided by Bioconductors `multtest` package.

3.1 Definition of the sample groups

The test statistics are performed on the expression values of the single signal channels of the arrays. The group 0 consists of the following signal channels:

- DE Green: Group 0, pair: 2
- EF Red: Group 0, pair: 1

The group 1

- EF Green: Group 1, pair: 1
- FG Red: Group 1, pair: 2

The following arrays / signal channels where not assigned to any one of the groups:

- AB Red: skipped
- AB Green: skipped

- BC Red: skipped
- BC Green: skipped
- CD Red: skipped
- CD Green: skipped
- DE Red: skipped
- FG Green: skipped
- GH Red: skipped
- GH Green: skipped
- HA Red: skipped
- HA Green: skipped

```
> library(multtest)
> library(RColorBrewer)
> source("utils.R")
> if (!exists("Eset", envir = globalenv())) {
+   Eset <- newMadbSet(Slides.norm)
+ }
```

3.2 Prefiltering of the data

To alleviate the loss of power from the formidable multiplicity of gene–by–gene hypothesis testing that is common to microarray experiments, a non–specific prefiltering should be carried out. Non–specific means without reference to the group the samples are into. The aim of the prefiltering step is to remove from consideration that set of genes that are not differentially expressed under any comparison. In figure 3.3 the standard deviation (in log2 scale) of each gene across all samples is plotted on the y axis against the mean of each gene across all samples (x-axis). The scatterplot of these values versus each other allows to visually verify whether there is a dependence of the standard deviation (or variance) on the mean. The red dots depict the running median estimator. If there is no variance-mean dependence, then the line formed by the red dots should be approximately horizontal. In such a case a prefiltering that bases solely on the variance can be performed.

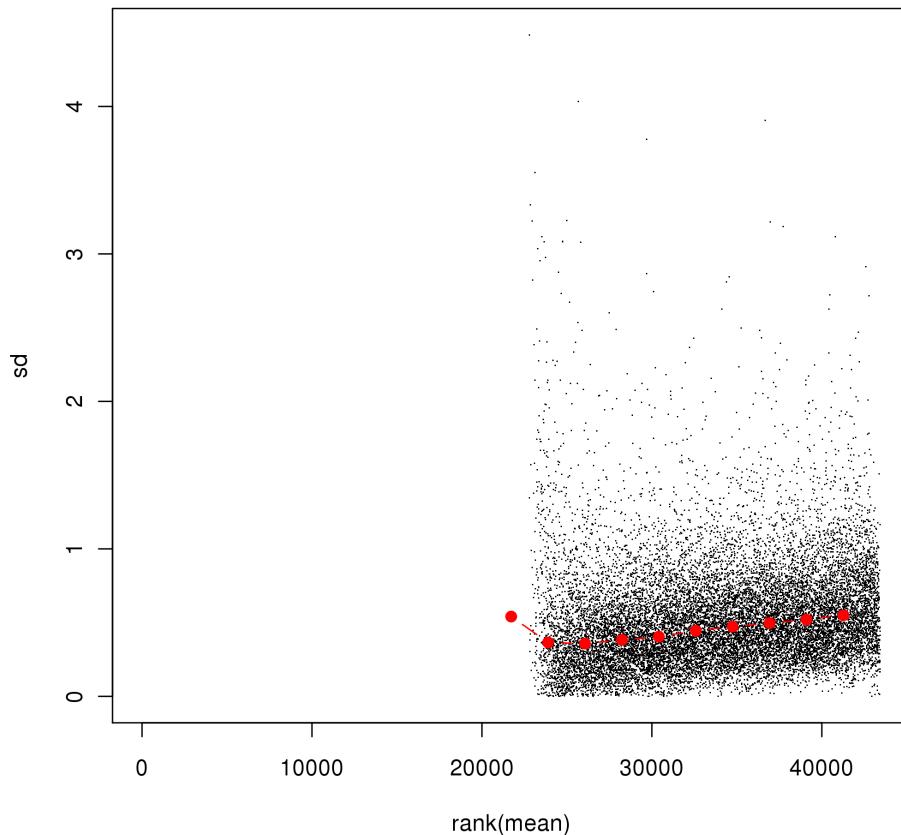


Figure 3.1: Mean vs standard deviation (in log2 scale) plot of the data. The red dots depict the running median estimator.

Using the 40% of the genes with the biggest variance over the samples. These genes have a standard deviation bigger than the one represented by the horizontal red line in figure 3.2.

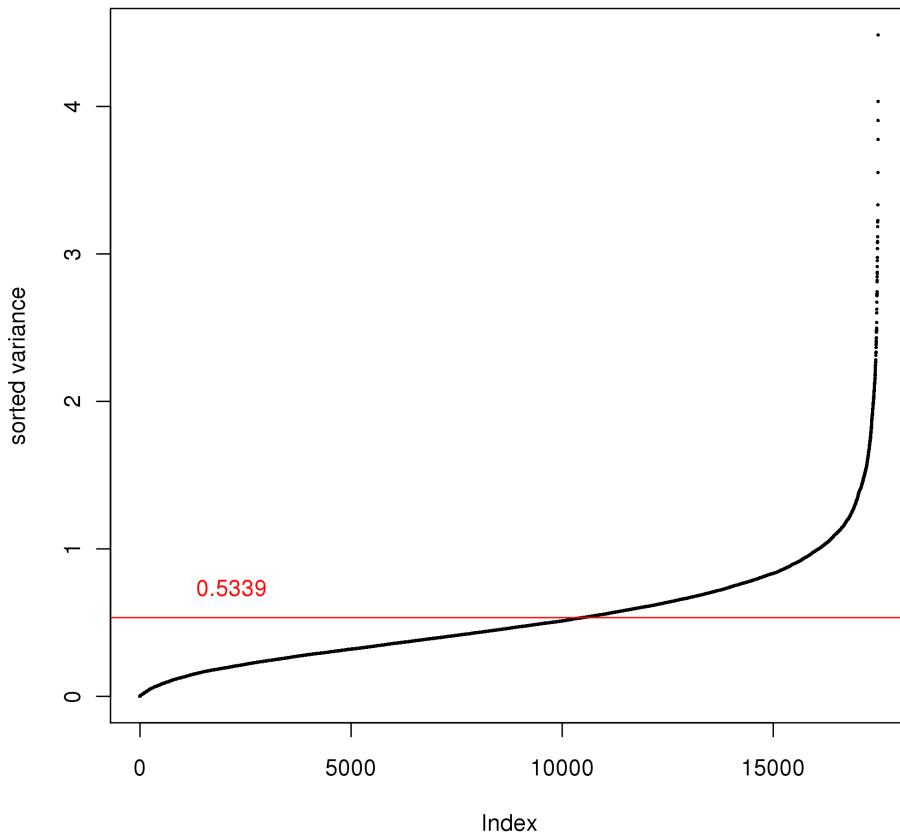


Figure 3.2: Sorted standard deviation of all genes across all samples. 40% of the genes have a larger variance (standard deviation) than the variance represented by the red horizontal line.

```
> Eset.filtered <- filterOnVariance(Eset, variance = 0.6, array.names = c("DE Green",
+ "EF Red", "EF Green", "FG Red"))

6996 features out of 43441 have a sd bigger than 0.5338951
```

3.3 Calculating the raw p values

Based on the selected test statistics p-values are calculated that give information about how significantly a gene is differentially expressed between the two groups. Genes that have not at least one value within each group, that was not flagged by the scanning software as a bad spot will be removed automatically from the further analysis.

```
> Classlabels <- c(0, 0, 1, 1)
> Cols <- c("DE Green", "EF Red", "EF Green", "FG Red")
> Data <- exprs(Eset.filtered)[, Cols]
```

3.4 Correcting the p values for multiple testing

```
> if (!.is.log(Data)) {  
+   Data <- log2(Data)  
+ }  
> rownames(Data) <- as.character(1:nrow(Data))  
> Excluded.genes <- excludeFromTest(Data, classlabels = Classlabels,  
+   weights = getWeights(Eset.filtered)[, Cols])
```

From the 6996 genes in the experiment 1712 were excluded from the further analysis due to the restriction that at least 2 gene per sample group should not be flagged by the scanning software as a bad spot.

Using paired *moderated t-statistics* provided by the *limma* package to calculate the raw p values.

```
> library(limma)  
> if (!.is.log(Data)) {  
+   Data <- log2(Data)  
+ }  
> design.samples <- factor(c(2, 1, 1, 2))  
> design.assignment <- factor(ifelse(Classlabels == 1, "sample",  
+   "ref"))  
> pdata <- 1:length(Classlabels)  
> names(pdata) <- colnames(Data)  
> design <- model.matrix(~design.samples + design.assignment, data = data.frame(pdata))  
> Fit <- lmFit(Data[!Excluded.genes, ], design)  
> Fit <- eBayes(Fit)  
> PValues <- as.matrix(Fit$p.value[, "design.assignment.sample"])  
> colnames(PValues) <- "rawP"
```

3.4 Correcting the p values for multiple testing

Microarray experiments generate large multiplicity problems in which thousands of hypothesis (is gene x differentially expressed between the two groups) are tested simultaneously. To correct for false positive (type I errors) and false negative (type 2) errors that occur in such a setting, different approaches have been developed. The simplest one is the *Bonferroni* adjustment method, that multiplies the raw p value with the number of hypothesis tested in the setting. For more information about the methods available please refer to the publication from Sandrine Dudoit *Multiple Hypothesis Testing in Microarray Experiments*.

```
> AdjP <- mt.rawp2adjp(PValues[, "rawP"], proc = c("BH"))  
> AdjP.ordered <- AdjP$adjp[order(AdjP$index), ]  
> p.idx <- AdjP$index  
> if (!exists("p.idx", envir = globalenv())) {  
+   p.idx <- order(as.numeric(PValues[, "rawP"]))  
+ }
```

The p-values adjusted with the various adjustment methods are plotted in figure 3.3 and 3.4. The plots of the sorted raw and adjusted p-values are a helpful tool for the decision of a cut-off value for significance. Significantly differentially expressed genes can be defined by using an appropriate combination of number of genes to follow up and tolerable false positive rate. Descriptions for the various abbreviations: *rawp*: unadjusted p-values, *Bonferroni*: Bonferroni adjusted p-values (strong control of the FWER (Family Wise Error Rate, the probability of at least one false positive)), *SidakSS*: Sidak's single step method

3.4 Correcting the p values for multiple testing

adjusted p-values (strong control of the FWER), *SidakSD*: Sidak's step down method adjusted p-values (strong control of the FWER), *Holm*: p-values adjusted using the method from Holm (strong control of the FWER), *Hochberg*: p-values adjusted using the Hochberg method (strong control of the FWER), *BH*: p-values adjusted using the method proposed by Benjamini and Hochberg (strong control of the FDR (False Discovery Rate, expected proportion of false positives among the rejected hypothesis)), *BY*: p-values adjusted using the method from Benjamini and Yekutieli (strong control of the FDR).

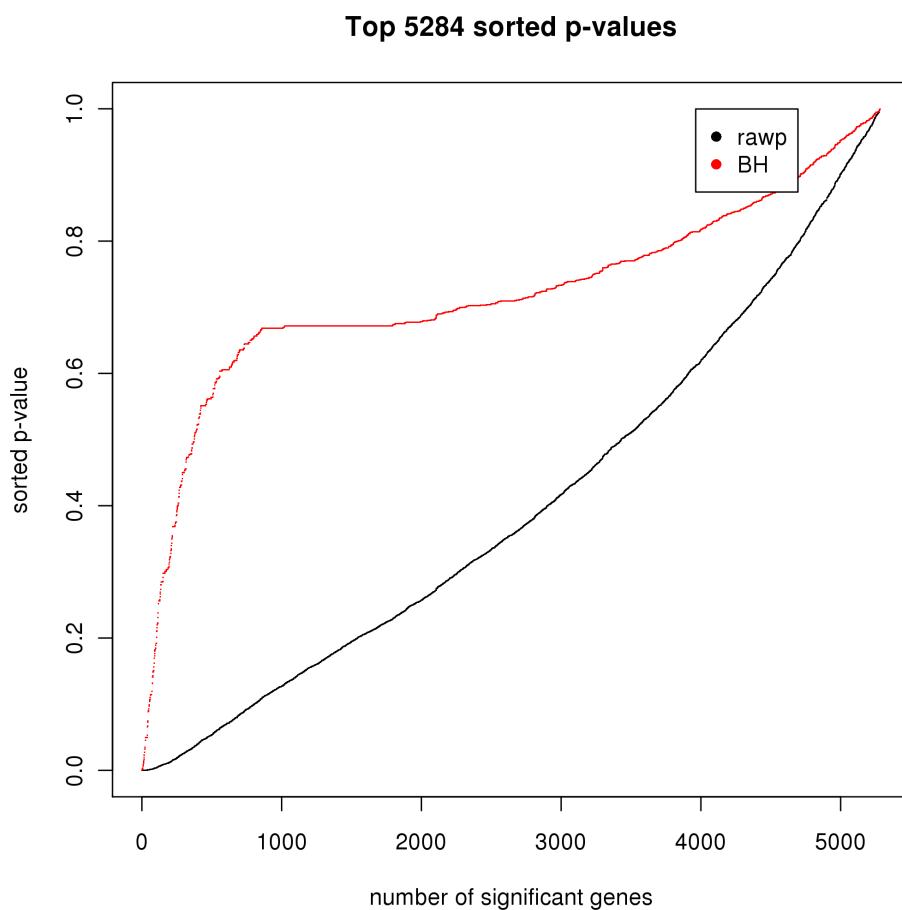


Figure 3.3: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

3.4 Correcting the p values for multipletesting differentially expressed genes using test statistics

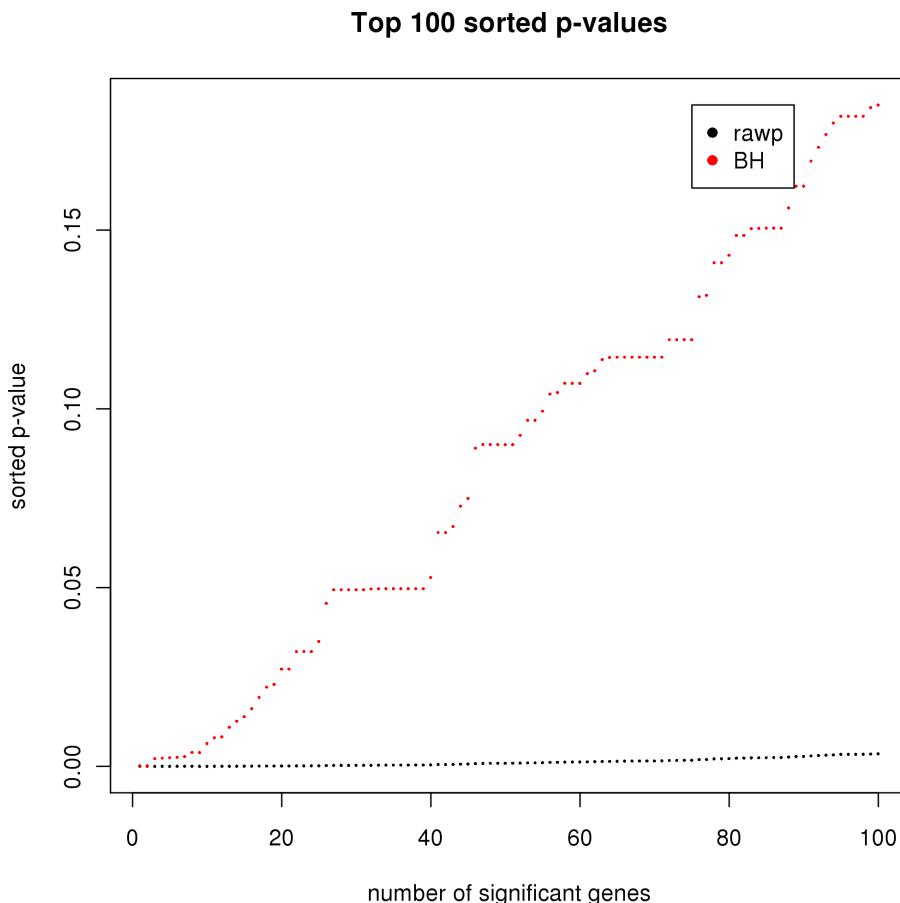


Figure 3.4: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

```
> Filename <- checkExistantFile("PR39B29.txt")
```

A table containing all calculated p values (raw p values and corrected p values) is saved as tabulator delimited text file to the file: *PR39B29.txt*; Calculating average regulation (M) and average expression (A) values between the two groups. The average M value for each gene is calculated by subtracting the average expression value of the gene in group 0 from the average expression value of the gene in group 1 (so a mean M of 1 means a two fold increase in the expression level of the gene in group 1 compared to the expression level in group 0).

The average MA plot is drawn using M and A values that are calculated from the average expression values of each gene in each sample group. To calculate the average ,the mean function is used.

```
> library(geneplotter)
> MAColor <- densCols(M, A, colramp = colorRampPalette(rev(brewer.pal(9,
+      "Blues"))[2:9]))
```

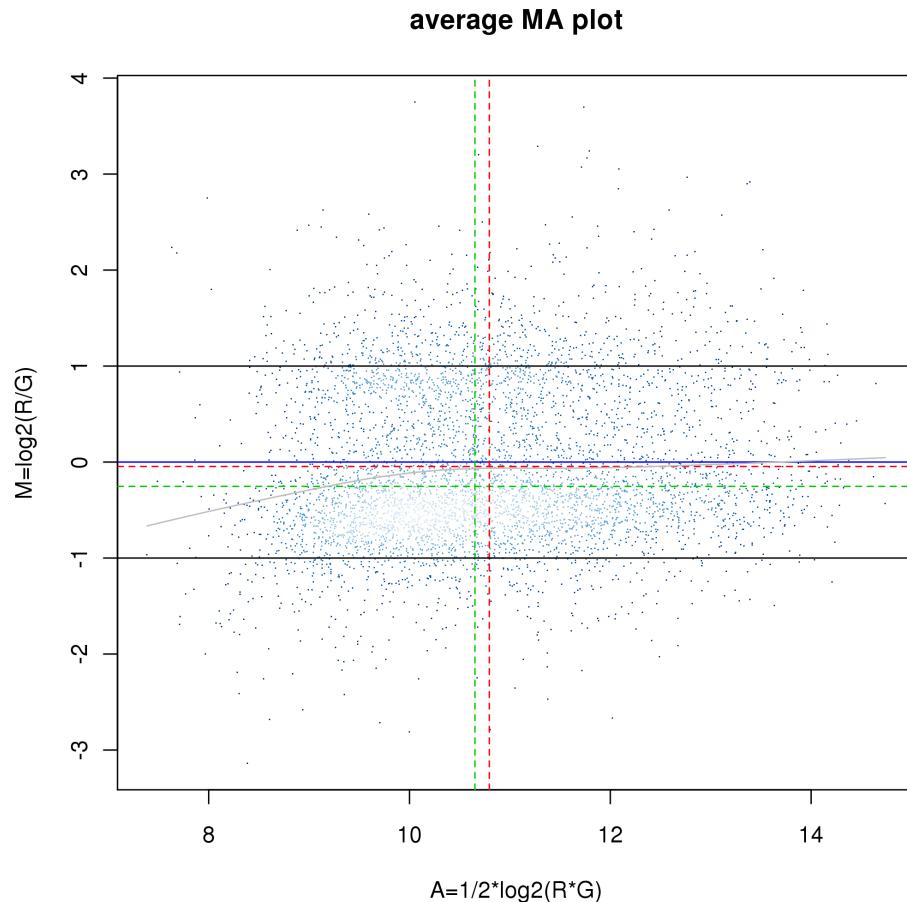


Figure 3.5: MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.

In the volcano plot the p values are scattered against the regulation values (average M values). The volcano plot in figure 6.6 represents the average M value per gene and the according raw p value calculated using the selected test statistics. The most interesting genes would be those that have both small p values and big average M values.

3.4 Correcting the p values for multiple testing differentially expressed genes using test statistics

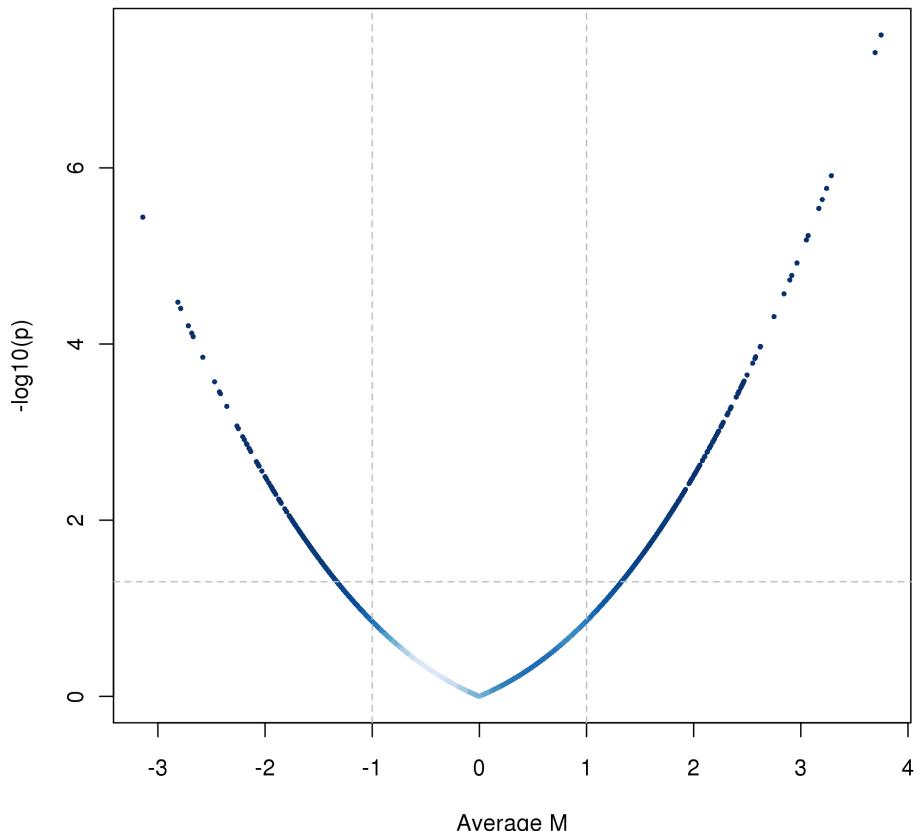


Figure 3.6: Volcano plot scattering the average M values (x axis) against the raw p values (y axis, $-\log_{10}$ scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

Volcano plot of the average M values against the p values corrected for multiple testing using the method proposed by Benjamini and Hochberg (figure 6.7).

3.4 Correcting the p values for multipletesting differentially expressed genes using test statistics

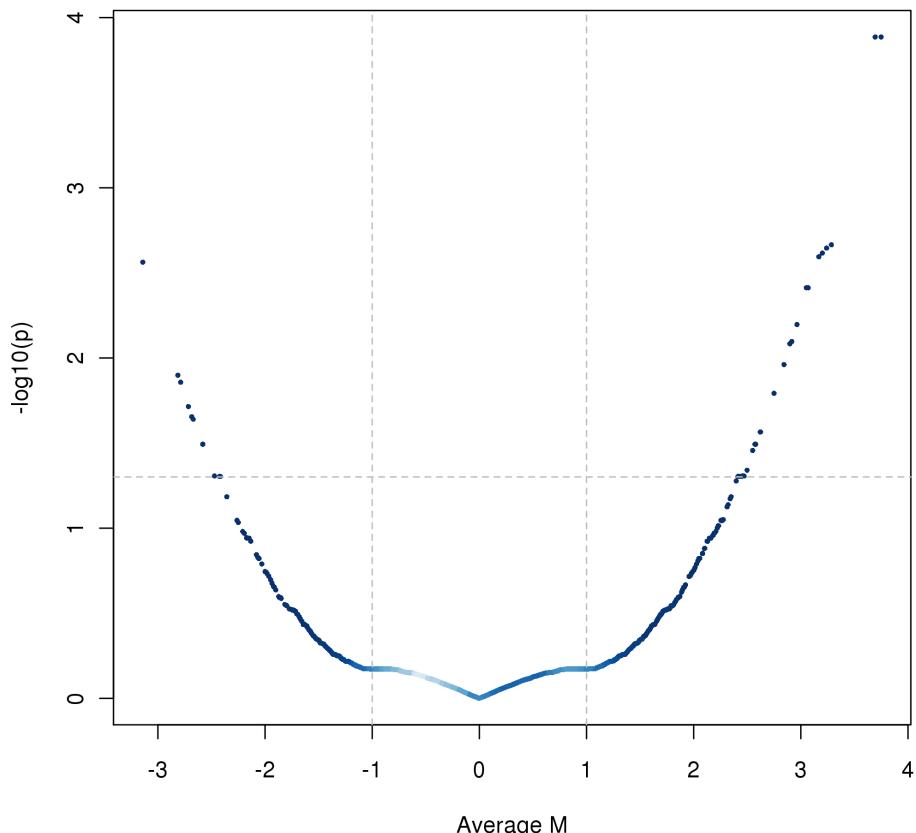


Figure 3.7: Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, $-\log_{10}$ scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

The p values together with the data on which the test statistic was calculated of the 100 genes with the smallest p values is saved to the file : *PR39B29-top100.txt*

Chapter 4

Determining differentially expressed genes using test statistics

Statistical tests are used in this chapter to define genes that are differentially expressed between two groups in this micro array experiment. Statistical tests allow to find genes, that show different expression levels between two sample groups and small alterations in expression levels within each group. The statistical tests used in this analysis are mainly provided by Bioconductors `multtest` package.

4.1 Definition of the sample groups

The test statistics are performed on the expression values of the single signal channels of the arrays. The group 0 consists of the following signal channels:

- FG Green: Group 0, pair: 2
- GH Red: Group 0, pair: 1

The group 1

- GH Green: Group 1, pair: 1
- HA Red: Group 1, pair: 2

The following arrays / signal channels where not assigned to any one of the groups:

- AB Red: skipped
- AB Green: skipped

- BC Red: skipped
- BC Green: skipped
- CD Red: skipped
- CD Green: skipped
- DE Red: skipped
- DE Green: skipped
- EF Red: skipped
- EF Green: skipped
- FG Red: skipped
- HA Green: skipped

```
> library(multtest)
> library(RColorBrewer)
> source("utils.R")
> if (!exists("Eset", envir = globalenv())) {
+   Eset <- newMadbSet(Slides.norm)
+ }
```

4.2 Prefiltering of the data

To alleviate the loss of power from the formidable multiplicity of gene–by–gene hypothesis testing that is common to microarray experiments, a non–specific prefiltering should be carried out. Non–specific means without reference to the group the samples are into. The aim of the prefiltering step is to remove from consideration that set of genes that are not differentially expressed under any comparison. In figure 4.3 the standard deviation (in log2 scale) of each gene across all samples is plotted on the y axis against the mean of each gene across all samples (x-axis). The scatterplot of these values versus each other allows to visually verify whether there is a dependence of the standard deviation (or variance) on the mean. The red dots depict the running median estimator. If there is no variance-mean dependence, then the line formed by the red dots should be approximately horizontal. In such a case a prefiltering that bases solely on the variance can be performed.

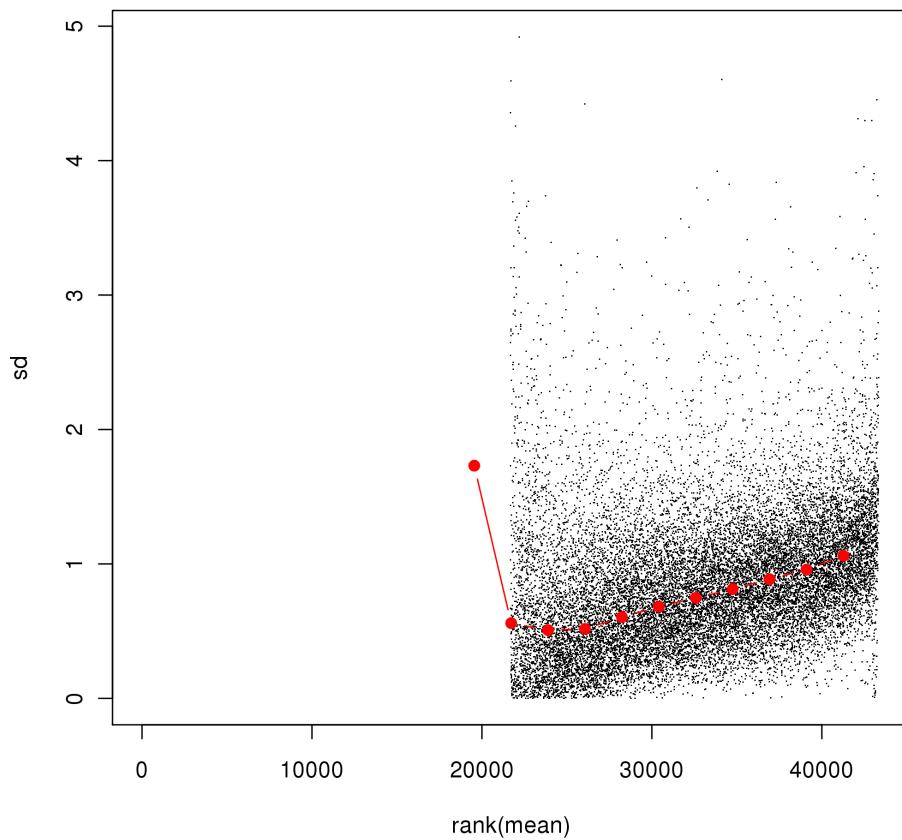


Figure 4.1: Mean vs standard deviation (in log2 scale) plot of the data. The red dots depict the running median estimator.

Using the 40% of the genes with the biggest variance over the samples. These genes have a standard deviation bigger than the one represented by the horizontal red line in figure 4.2.

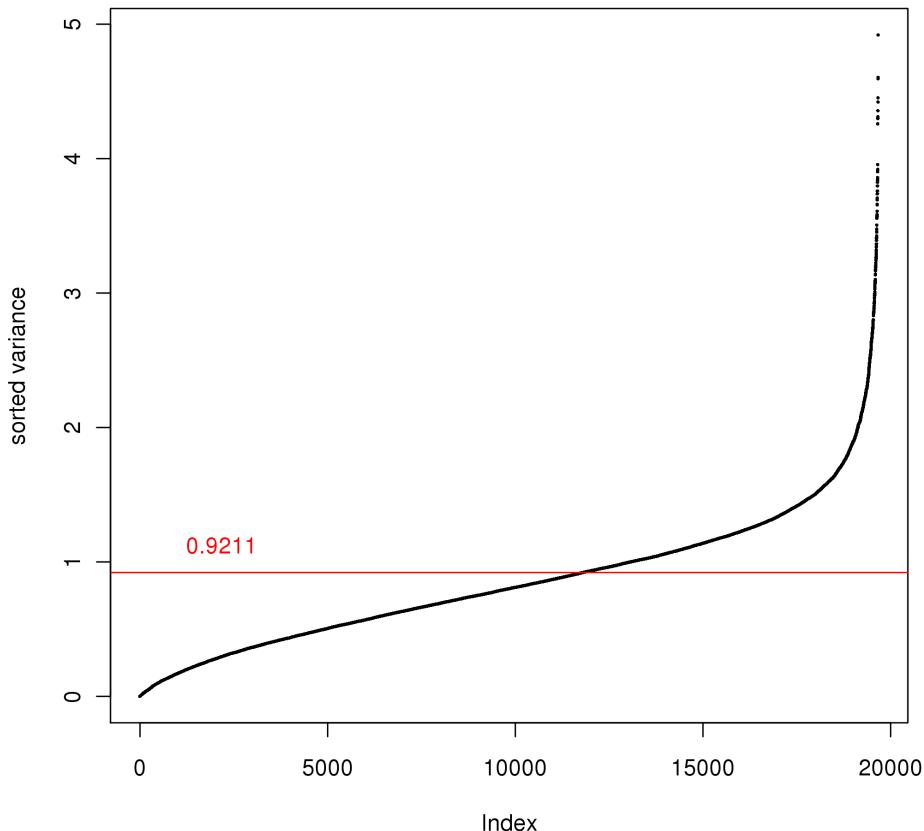


Figure 4.2: Sorted standard deviation of all genes across all samples. 40% of the genes have a larger variance (standard deviation) than the variance represented by the red horizontal line.

```
> Eset.filtered <- filterOnVariance(Eset, variance = 0.6, array.names = c("FG Green",
+ "GH Red", "GH Green", "HA Red"))

7869 features out of 43441 have a sd bigger than 0.9211216
```

4.3 Calculating the raw p values

Based on the selected test statistics p-values are calculated that give information about how significantly a gene is differentially expressed between the two groups. Genes that have not at least one value within each group, that was not flagged by the scanning software as a bad spot will be removed automatically from the further analysis.

```
> Classlabels <- c(0, 0, 1, 1)
> Cols <- c("FG Green", "GH Red", "GH Green", "HA Red")
> Data <- exprs(Eset.filtered)[, Cols]
```

4.4 Correcting the p values for multiple testing

```
> if (!.is.log(Data)) {
+   Data <- log2(Data)
+ }
> rownames(Data) <- as.character(1:nrow(Data))
> Excluded.genes <- excludeFromTest(Data, classlabels = Classlabels,
+   weights = getWeights(Eset.filtered)[, Cols])
```

From the 7869 genes in the experiment 2106 were excluded from the further analysis due to the restriction that at least 2 gene per sample group should not be flagged by the scanning software as a bad spot.

Using paired *moderated t-statistics* provided by the *limma* package to calculate the raw p values.

```
> library(limma)
> if (!.is.log(Data)) {
+   Data <- log2(Data)
+ }
> design.samples <- factor(c(2, 1, 1, 2))
> design.assignment <- factor(ifelse(Classlabels == 1, "sample",
+   "ref"))
> pdata <- 1:length(Classlabels)
> names(pdata) <- colnames(Data)
> design <- model.matrix(~design.samples + design.assignment, data = data.frame(pdata))
> Fit <- lmFit(Data[!Excluded.genes, ], design)
> Fit <- eBayes(Fit)
> PValues <- as.matrix(Fit$p.value[, "design.assignment.sample"])
> colnames(PValues) <- "rawp"
```

4.4 Correcting the p values for multiple testing

Microarray experiments generate large multiplicity problems in which thousands of hypothesis (is gene x differentially expressed between the two groups) are tested simultaneously. To correct for false positive (type I errors) and false negative (type 2) errors that occur in such a setting, different approaches have been developed. The simplest one is the *Bonferroni* adjustment method, that multiplies the raw p value with the number of hypothesis tested in the setting. For more information about the methods available please refer to the publication from Sandrine Dudoit *Multiple Hypothesis Testing in Microarray Experiments*.

```
> AdjP <- mt.rawp2adjp(PValues[, "rawp"], proc = c("BH"))
> AdjP.ordered <- AdjP$adjp[order(AdjP$index), ]
> p.idx <- AdjP$index
> if (!exists("p.idx", envir = globalenv())) {
+   p.idx <- order(as.numeric(PValues[, "rawp"]))
+ }
```

The p-values adjusted with the various adjustment methods are plotted in figure 4.3 and 4.4. The plots of the sorted raw and adjusted p-values are a helpful tool for the decision of a cut-off value for significance. Significantly differentially expressed genes can be defined by using an appropriate combination of number of genes to follow up and tolerable false positive rate. Descriptions for the various abbreviations: *rawp*: unadjusted p-values, *Bonferroni*: Bonferroni adjusted p-values (strong control of the FWER (Family Wise Error Rate, the probability of at least one false positive)), *SidakSS*: Sidak's single step method

4.4 Correcting the p values for multiple testing

adjusted p-values (strong control of the FWER), *SidakSD*: Sidak's step down method adjusted p-values (strong control of the FWER), *Holm*: p-values adjusted using the method from Holm (strong control of the FWER), *Hochberg*: p-values adjusted using the Hochberg method (strong control of the FWER), *BH*: p-values adjusted using the method proposed by Benjamini and Hochberg (strong control of the FDR (False Discovery Rate, expected proportion of false positives among the rejected hypothesis)), *BY*: p-values adjusted using the method from Benjamini and Yekutieli (strong control of the FDR).

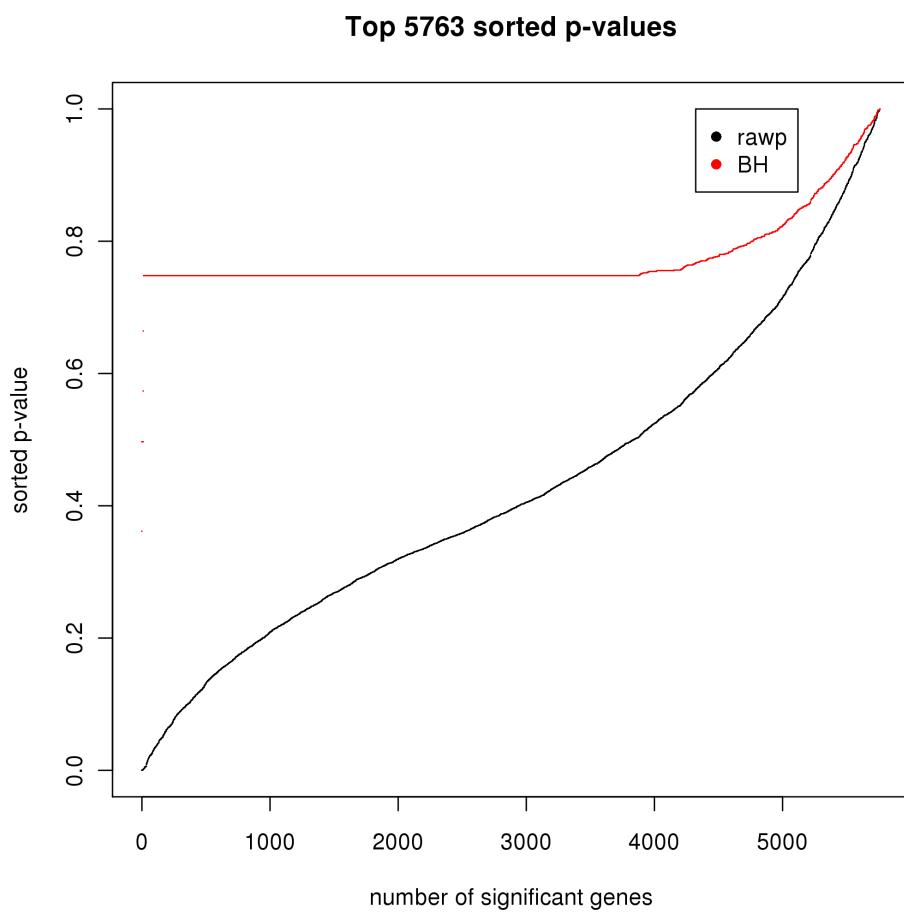


Figure 4.3: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

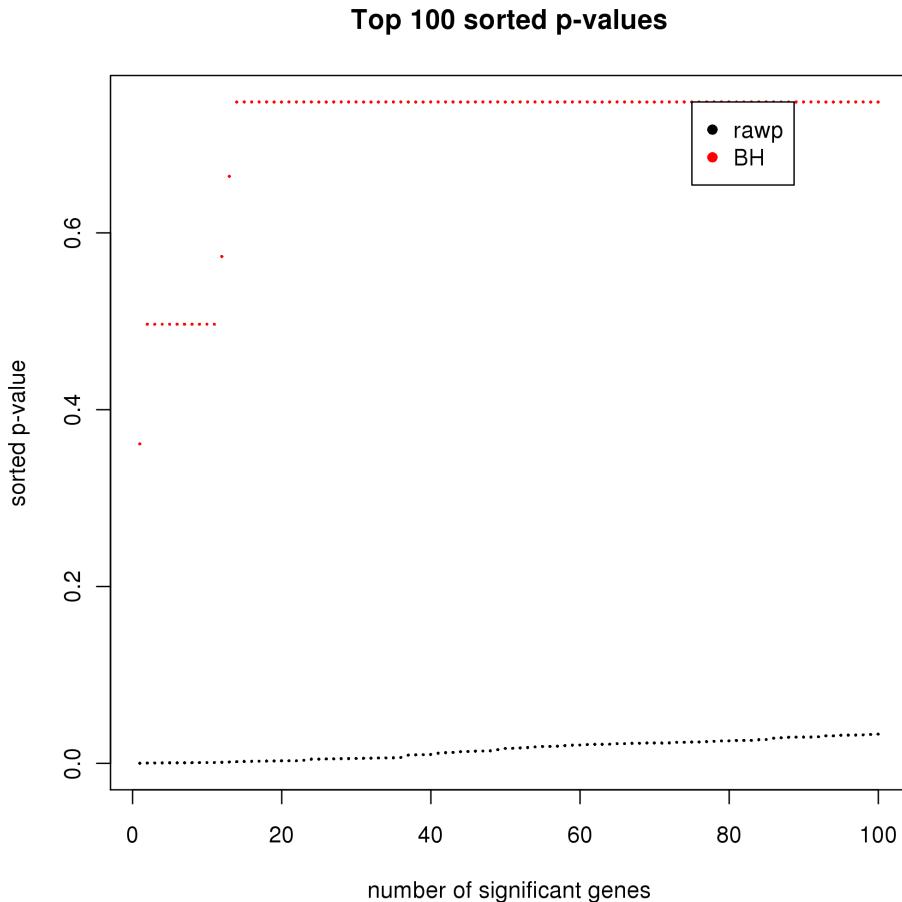


Figure 4.4: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

```
> Filename <- checkExistantFile("CODISCO.txt")
```

A table containing all calculated p values (raw p values and corrected p values) is saved as tabulator delimited text file to the file: *CODISCO.txt*; Calculating average regulation (M) and average expression (A) values between the two groups. The average M value for each gene is calculated by subtracting the average expression value of the gene in group 0 from the average expression value of the gene in group 1 (so a mean M of 1 means a two fold increase in the expression level of the gene in group 1 compared to the expression level in group 0).

The average MA plot is drawn using M and A values that are calculated from the average expression values of each gene in each sample group. To calculate the average ,the mean function is used.

```
> library(geneplotter)
> MAColor <- densCols(M, A, colramp = colorRampPalette(rev(brewer.pal(9,
+ "Blues"))[2:9])))
```

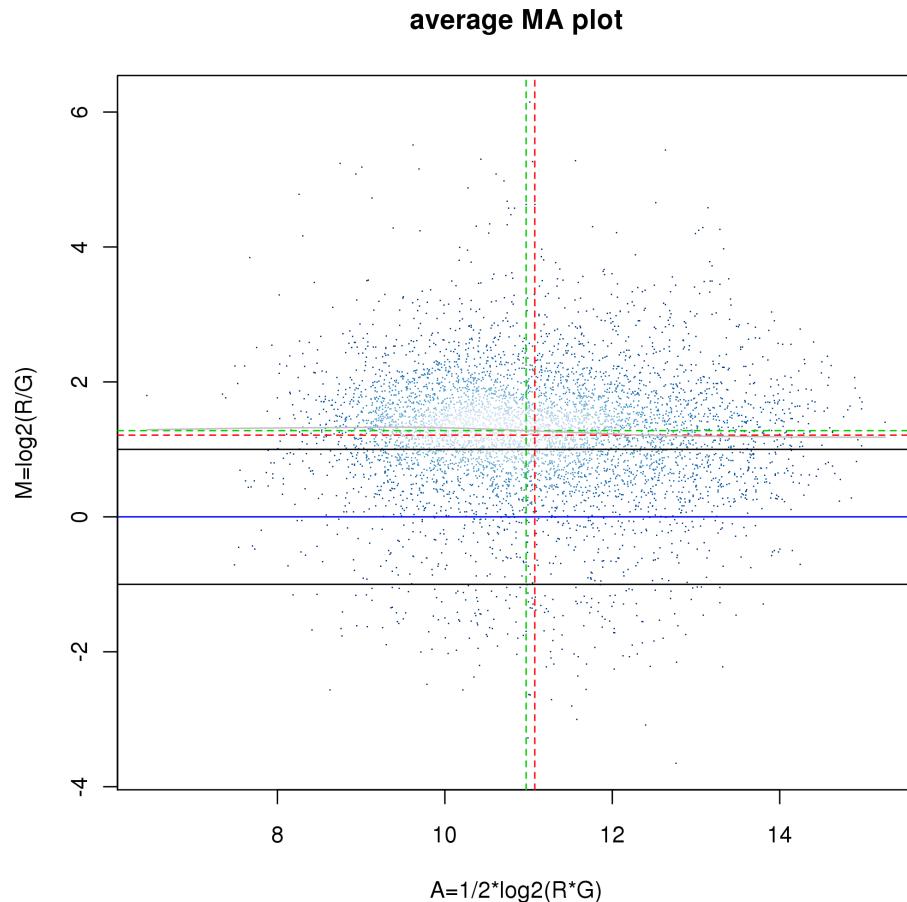


Figure 4.5: MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.

In the volcano plot the p values are scattered against the regulation values (average M values). The volcano plot in figure 6.6 represents the average M value per gene and the according raw p value calculated using the selected test statistics. The most interesting genes would be those that have both small p values and big average M values.

4.4 Correcting the p values for multiple testing differentially expressed genes using test statistics

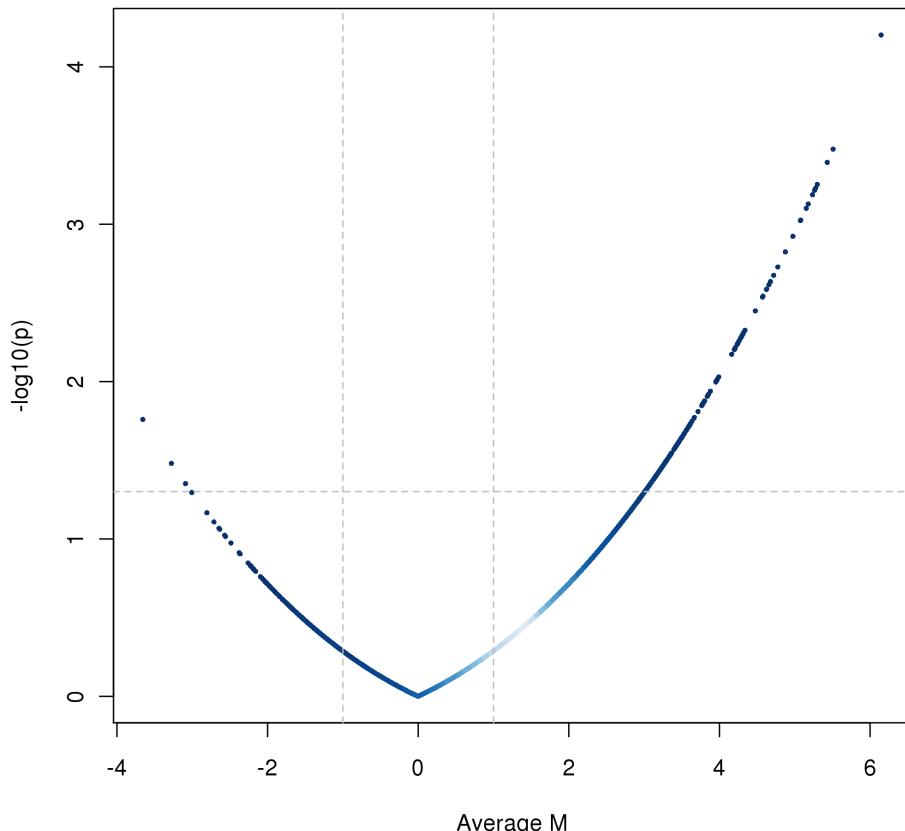


Figure 4.6: Volcano plot scattering the average M values (x axis) against the raw p values (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

Volcano plot of the average M values against the p values corrected for multiple testing using the method proposed by Benjamini and Hochberg (figure 6.7).

4.4 Correcting the p values for multip~~testing~~detecting differentially expressed genes using test statistics

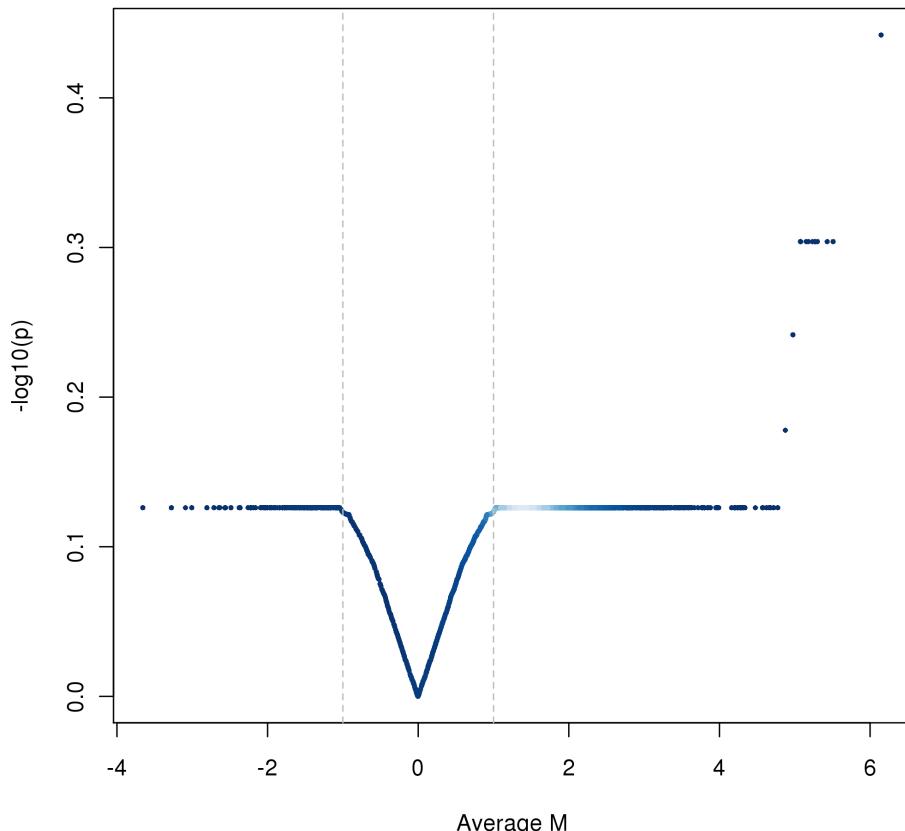


Figure 4.7: Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, $-\log_{10}$ scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

The p values together with the data on which the test statistic was calculated of the 100 genes with the smallest p values is saved to the file : *CODISCO-top100.txt*

Chapter 5

Determining differentially expressed genes using test statistics

Statistical tests are used in this chapter to define genes that are differentially expressed between two groups in this micro array experiment. Statistical tests allow to find genes, that show different expression levels between two sample groups and small alterations in expression levels within each group. The statistical tests used in this analysis are mainly provided by Bioconductors `multtest` package.

5.1 Definition of the sample groups

The test statistics are performed on the expression values of the single signal channels of the arrays. The group 0 consists of the following signal channels:

- BC Green: Group 0, pair: 2
- CD Red: Group 0, pair: 1

The group 1

- CD Green: Group 1, pair: 1
- DE Red: Group 1, pair: 2

The following arrays / signal channels were not assigned to any one of the groups:

- AB Red: skipped
- AB Green: skipped

- BC Red: skipped
- DE Green: skipped
- EF Red: skipped
- EF Green: skipped
- FG Red: skipped
- FG Green: skipped
- GH Red: skipped
- GH Green: skipped
- HA Red: skipped
- HA Green: skipped

```
> library(multtest)
> library(RColorBrewer)
> source("utils.R")
> if (!exists("Eset", envir = globalenv())) {
+   Eset <- newMadbSet(Slides.norm)
+ }
```

5.2 Prefiltering of the data

To alleviate the loss of power from the formidable multiplicity of gene–by–gene hypothesis testing that is common to microarray experiments, a non–specific prefiltering should be carried out. Non–specific means without reference to the group the samples are into. The aim of the prefiltering step is to remove from consideration that set of genes that are not differentially expressed under any comparison. In figure 5.1 the standard deviation (in log2 scale) of each gene across all samples is plotted on the y axis against the mean of each gene across all samples (x-axis). The scatterplot of these values versus each other allows to visually verify whether there is a dependence of the standard deviation (or variance) on the mean. The red dots depict the running median estimator. If there is no variance-mean dependence, then the line formed by the red dots should be approximately horizontal. In such a case a prefiltering that bases solely on the variance can be performed.

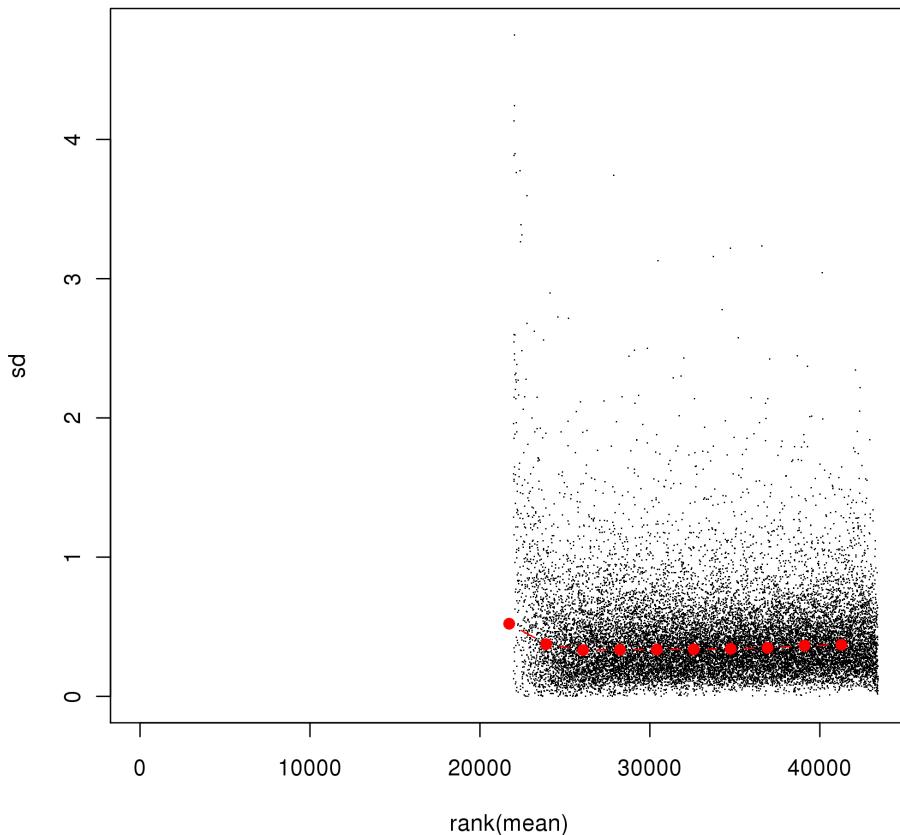


Figure 5.1: Mean vs standard deviation (in log2 scale) plot of the data. The red dots depict the running median estimator.

Using the 40% of the genes with the biggest variance over the samples. These genes have a standard deviation bigger than the one represented by the horizontal red line in figure 5.2.

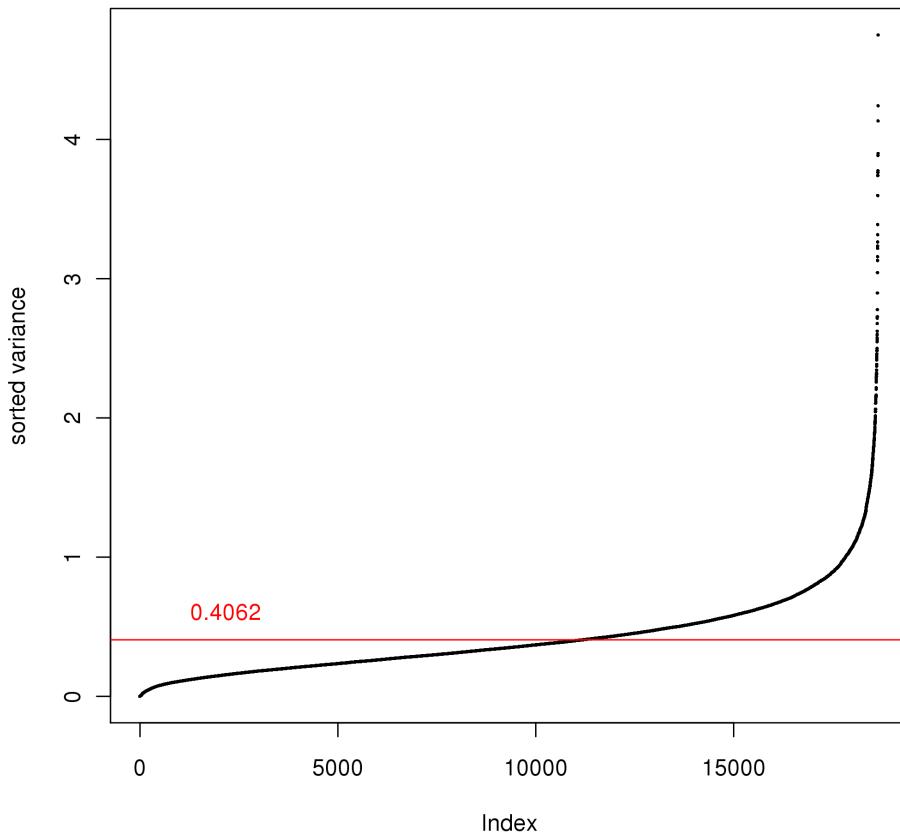


Figure 5.2: Sorted standard deviation of all genes across all samples. 40% of the genes have a larger variance (standard deviation) than the variance represented by the red horizontal line.

```
> Eset.filtered <- filterOnVariance(Eset, variance = 0.6, array.names = c("BC Green",
+ "CD Red", "CD Green", "DE Red"))

7462 features out of 43441 have a sd bigger than 0.4061636
```

5.3 Calculating the raw p values

Based on the selected test statistics p-values are calculated that give information about how significantly a gene is differentially expressed between the two groups. Genes that have not at least one value within each group, that was not flagged by the scanning software as a bad spot will be removed automatically from the further analysis.

```
> Classlabels <- c(0, 0, 1, 1)
> Cols <- c("BC Green", "CD Red", "CD Green", "DE Red")
> Data <- exprs(Eset.filtered)[, Cols]
```

5.4 Correcting the p values for multiple testing

```
> if (!.is.log(Data)) {
+   Data <- log2(Data)
+ }
> rownames(Data) <- as.character(1:nrow(Data))
> Excluded.genes <- excludeFromTest(Data, classlabels = Classlabels,
+   weights = getWeights(Eset.filtered)[, Cols])
```

From the 7462 genes in the experiment 2201 were excluded from the further analysis due to the restriction that at least 2 gene per sample group should not be flagged by the scanning software as a bad spot.

Using paired *moderated t-statistics* provided by the *limma* package to calculate the raw p values.

```
> library(limma)
> if (!.is.log(Data)) {
+   Data <- log2(Data)
+ }
> design.samples <- factor(c(2, 1, 1, 2))
> design.assignment <- factor(ifelse(Classlabels == 1, "sample",
+   "ref"))
> pdata <- 1:length(Classlabels)
> names(pdata) <- colnames(Data)
> design <- model.matrix(~design.samples + design.assignment, data = data.frame(pdata))
> Fit <- lmFit(Data[!Excluded.genes, ], design)
> Fit <- eBayes(Fit)
> PValues <- as.matrix(Fit$p.value[, "design.assignment.sample"])
> colnames(PValues) <- "rawp"
```

5.4 Correcting the p values for multiple testing

Microarray experiments generate large multiplicity problems in which thousands of hypothesis (is gene x differentially expressed between the two groups) are tested simultaneously. To correct for false positive (type I errors) and false negative (type 2) errors that occur in such a setting, different approaches have been developed. The simplest one is the *Bonferroni* adjustment method, that multiplies the raw p value with the number of hypothesis tested in the setting. For more information about the methods available please refer to the publication from Sandrine Dudoit *Multiple Hypothesis Testing in Microarray Experiments*.

```
> AdjP <- mt.rawp2adjp(PValues[, "rawp"], proc = c("BH"))
> AdjP.ordered <- AdjP$adjp[order(AdjP$index), ]
> p.idx <- AdjP$index
> if (!exists("p.idx", envir = globalenv())) {
+   p.idx <- order(as.numeric(PValues[, "rawp"]))
+ }
```

The p-values adjusted with the various adjustment methods are plotted in figure 5.3 and 5.4. The plots of the sorted raw and adjusted p-values are a helpful tool for the decision of a cut-off value for significance. Significantly differentially expressed genes can be defined by using an appropriate combination of number of genes to follow up and tolerable false positive rate. Descriptions for the various abbreviations: *rawp*: unadjusted p-values, *Bonferroni*: Bonferroni adjusted p-values (strong control of the FWER (Family Wise Error Rate, the probability of at least one false positive)), *SidakSS*: Sidak's single step method

5.4 Correcting the p values for multiple testing

adjusted p-values (strong control of the FWER), *SidakSD*: Sidak's step down method adjusted p-values (strong control of the FWER), *Holm*: p-values adjusted using the method from Holm (strong control of the FWER), *Hochberg*: p-values adjusted using the Hochberg method (strong control of the FWER), *BH*: p-values adjusted using the method proposed by Benjamini and Hochberg (strong control of the FDR (False Discovery Rate, expected proportion of false positives among the rejected hypothesis)), *BY*: p-values adjusted using the method from Benjamini and Yekutieli (strong control of the FDR).

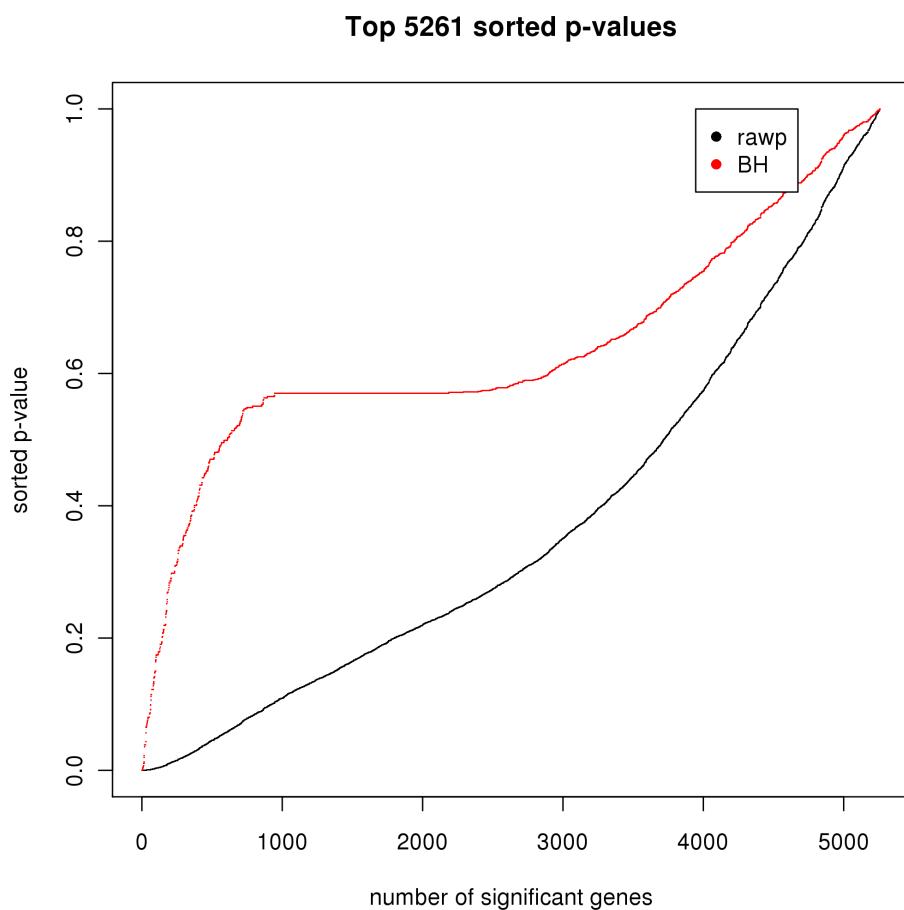


Figure 5.3: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

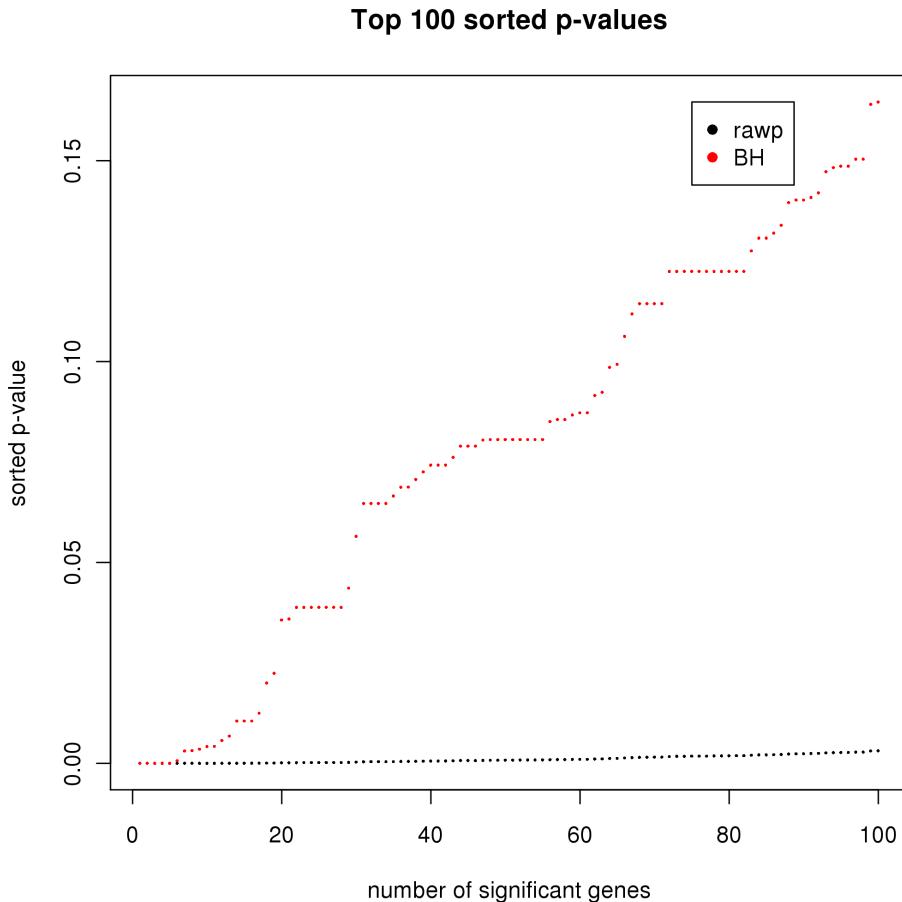


Figure 5.4: Plot of the sorted p -values. A description of the plot and the abbreviations is given in the text.

```
> Filename <- checkExistantFile("PICKER.txt")
```

A table containing all calculated p values (raw p values and corrected p values) is saved as tabulator delimited text file to the file: *PICKER.txt*; Calculating average regulation (M) and average expression (A) values between the two groups. The average M value for each gene is calculated by subtracting the average expression value of the gene in group 0 from the average expression value of the gene in group 1 (so a mean M of 1 means a two fold increase in the expression level of the gene in group 1 compared to the expression level in group 0).

The average MA plot is drawn using M and A values that are calculated from the average expression values of each gene in each sample group. To calculate the average ,the mean function is used.

```
> library(geneplotter)
> MAColor <- densCols(M, A, colramp = colorRampPalette(rev(brewer.pal(9,
+ "Blues"))[2:9]))
```

5.4 Correcting the p values for multiple testing

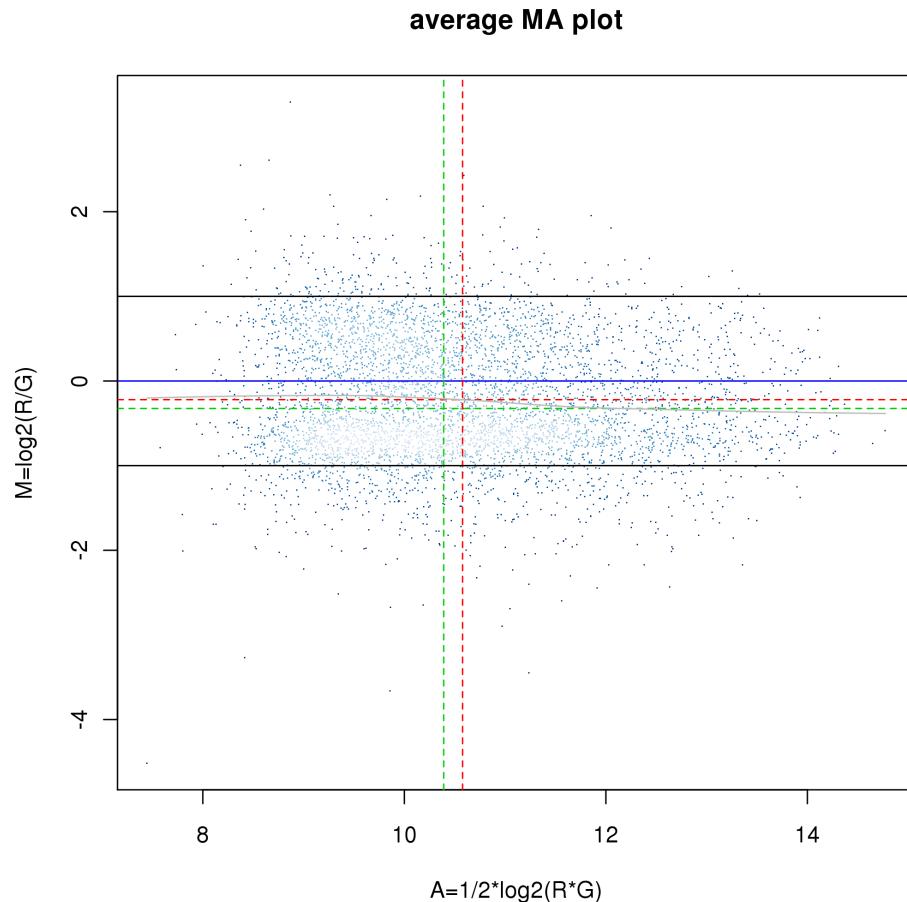


Figure 5.5: MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.

In the volcano plot the p values are scattered against the regulation values (average M values). The volcano plot in figure 6.6 represents the average M value per gene and the according raw p value calculated using the selected test statistics. The most interesting genes would be those that have both small p values and big average M values.

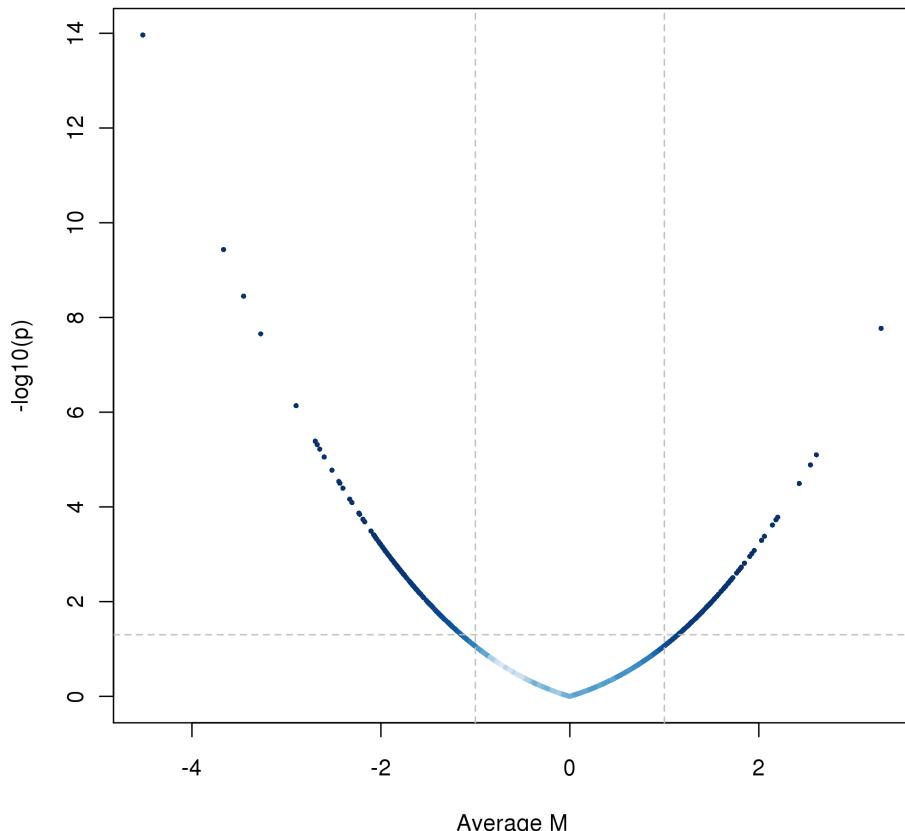


Figure 5.6: Volcano plot scattering the average M values (x axis) against the raw p values (y axis, $-\log_{10}$ scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

Volcano plot of the average M values against the p values corrected for multiple testing using the method proposed by Benjamini and Hochberg (figure 6.7).

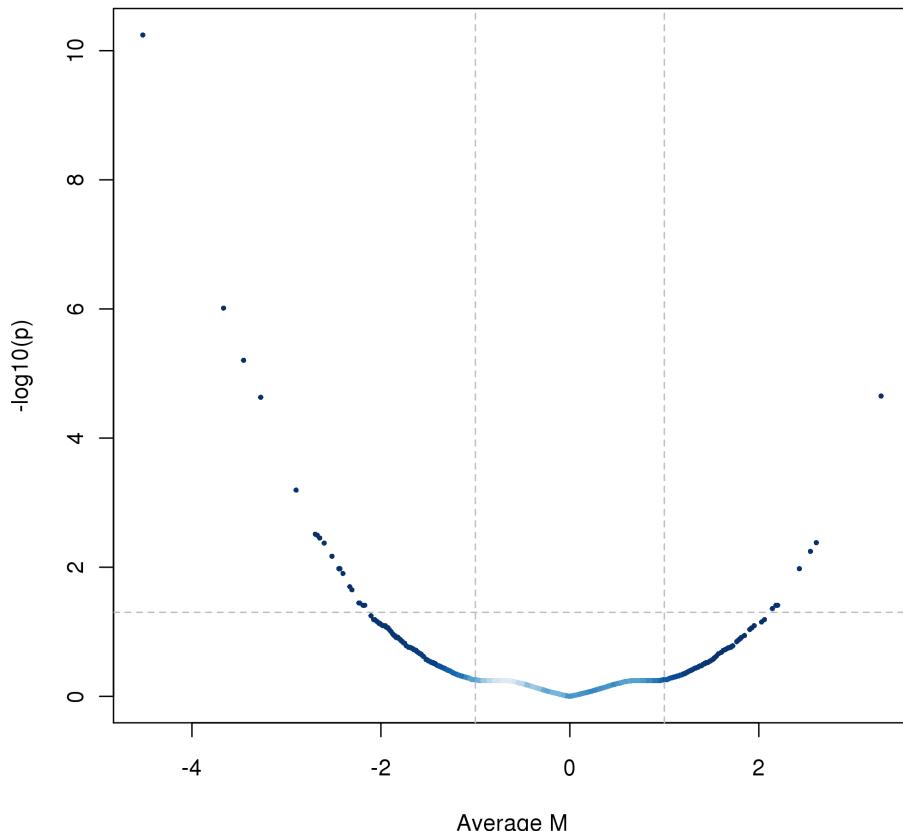


Figure 5.7: Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, $-\log_{10}$ scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

The p values together with the data on which the test statistic was calculated of the 100 genes with the smallest p values is saved to the file : *PICKER-top100.txt*

Chapter 6

Determining differentially expressed genes using test statistics

Statistical tests are used in this chapter to define genes that are differentially expressed between two groups in this micro array experiment. Statistical tests allow to find genes, that show different expression levels between two sample groups and small alterations in expression levels within each group. The statistical tests used in this analysis are mainly provided by Bioconductors `multtest` package.

6.1 Definition of the sample groups

The test statistics are performed on the expression values of the single signal channels of the arrays. The group 0 consists of the following signal channels:

- AB Red: Group 0, pair: 1
- HA Green: Group 0, pair: 2

The group 1

- AB Green: Group 1, pair: 1
- BC Red: Group 1, pair: 2

The following arrays / signal channels where not assigned to any one of the groups:

- BC Green: skipped
- CD Red: skipped

- CD Green: skipped
- DE Red: skipped
- DE Green: skipped
- EF Red: skipped
- EF Green: skipped
- FG Red: skipped
- FG Green: skipped
- GH Red: skipped
- GH Green: skipped
- HA Red: skipped

```
> library(multtest)
> library(RColorBrewer)
> source("utils.R")
> if (!exists("Eset", envir = globalenv())) {
+   Eset <- newMadbSet(Slides.norm)
+ }
```

6.2 Prefiltering of the data

To alleviate the loss of power from the formidable multiplicity of gene–by–gene hypothesis testing that is common to microarray experiments, a non–specific prefiltering should be carried out. Non–specific means without reference to the group the samples are into. The aim of the prefiltering step is to remove from consideration that set of genes that are not differentially expressed under any comparison. In figure 6.1 the standard deviation (in log2 scale) of each gene across all samples is plotted on the y axis against the mean of each gene across all samples (x-axis). The scatterplot of these values versus each other allows to visually verify whether there is a dependence of the standard deviation (or variance) on the mean. The red dots depict the running median estimator. If there is no variance-mean dependence, then the line formed by the red dots should be approximately horizontal. In such a case a prefiltering that bases solely on the variance can be performed.

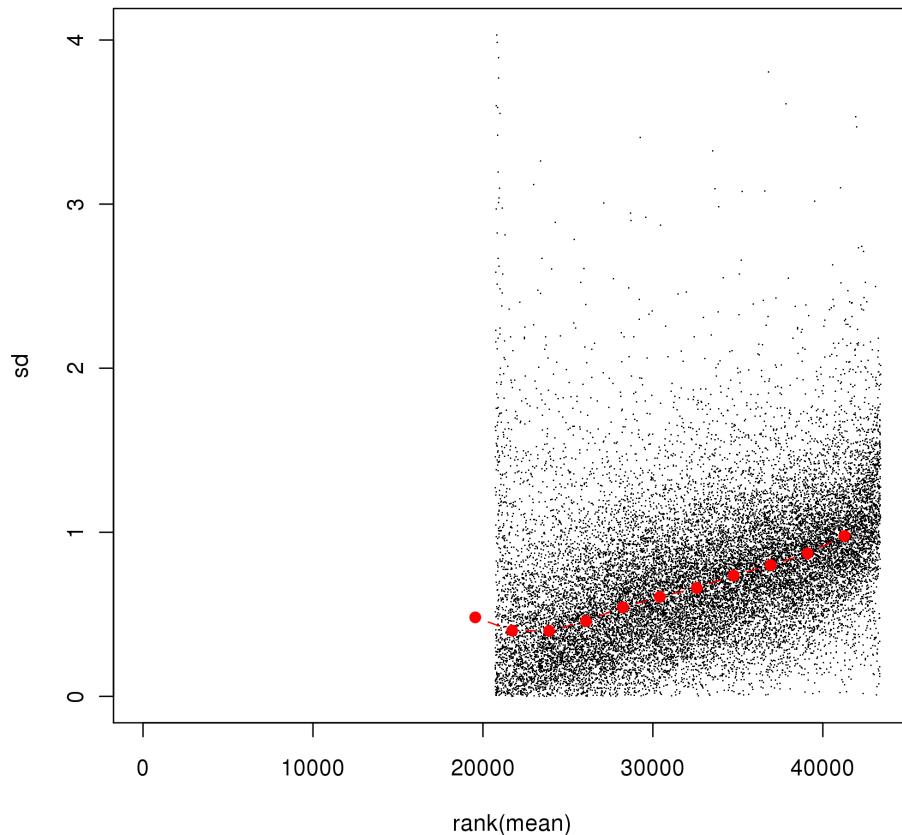


Figure 6.1: Mean vs standard deviation (in log2 scale) plot of the data. The red dots depict the running median estimator.

Using the 40% of the genes with the biggest variance over the samples. These genes have a standard deviation bigger than the one represented by the horizontal red line in figure 6.2.

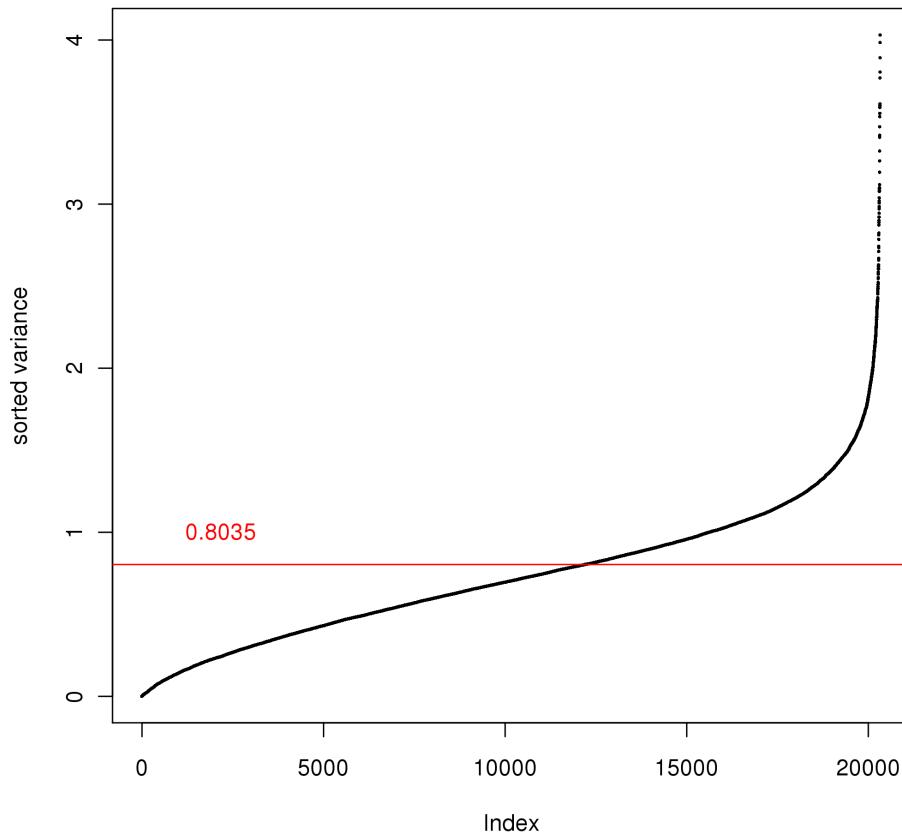


Figure 6.2: Sorted standard deviation of all genes across all samples. 40% of the genes have a larger variance (standard deviation) than the variance represented by the red horizontal line.

```
> Eset.filtered <- filterOnVariance(Eset, variance = 0.6, array.names = c("AB Red",
+ "AB Green", "BC Red", "HA Green"))

8133 features out of 43441 have a sd bigger than 0.803528
```

6.3 Calculating the raw p values

Based on the selected test statistics p-values are calculated that give information about how significantly a gene is differentially expressed between the two groups. Genes that have not at least one value within each group, that was not flagged by the scanning software as a bad spot will be removed automatically from the further analysis.

```
> Classlabels <- c(0, 1, 1, 0)
> Cols <- c("AB Red", "AB Green", "BC Red", "HA Green")
> Data <- exprs(Eset.filtered)[, Cols]
```

6.4 Correcting the p values for multiple testing

```
> if (!.is.log(Data)) {
+   Data <- log2(Data)
+ }
> rownames(Data) <- as.character(1:nrow(Data))
> Excluded.genes <- excludeFromTest(Data, classlabels = Classlabels,
+   weights = getWeights(Eset.filtered)[, Cols])
```

From the 8133 genes in the experiment 1751 were excluded from the further analysis due to the restriction that at least 2 gene per sample group should not be flagged by the scanning software as a bad spot.

Using paired *moderated t-statistics* provided by the *limma* package to calculate the raw p values.

```
> library(limma)
> if (!.is.log(Data)) {
+   Data <- log2(Data)
+ }
> design.samples <- factor(c(1, 1, 2, 2))
> design.assignment <- factor(ifelse(Classlabels == 1, "sample",
+   "ref"))
> pdata <- 1:length(Classlabels)
> names(pdata) <- colnames(Data)
> design <- model.matrix(~design.samples + design.assignment, data = data.frame(pdata))
> Fit <- lmFit(Data[!Excluded.genes, ], design)
> Fit <- eBayes(Fit)
> PValues <- as.matrix(Fit$p.value[, "design.assignment.sample"])
> colnames(PValues) <- "rawp"
```

6.4 Correcting the p values for multiple testing

Microarray experiments generate large multiplicity problems in which thousands of hypothesis (is gene x differentially expressed between the two groups) are tested simultaneously. To correct for false positive (type I errors) and false negative (type 2) errors that occur in such a setting, different approaches have been developed. The simplest one is the *Bonferroni* adjustment method, that multiplies the raw p value with the number of hypothesis tested in the setting. For more information about the methods available please refer to the publication from Sandrine Dudoit *Multiple Hypothesis Testing in Microarray Experiments*.

```
> AdjP <- mt.rawp2adjp(PValues[, "rawp"], proc = c("BH"))
> AdjP.ordered <- AdjP$adjp[order(AdjP$index), ]
> p.idx <- AdjP$index
> if (!exists("p.idx", envir = globalenv())) {
+   p.idx <- order(as.numeric(PValues[, "rawp"]))
+ }
```

The p-values adjusted with the various adjustment methods are plotted in figure 6.3 and 6.4. The plots of the sorted raw and adjusted p-values are a helpful tool for the decision of a cut-off value for significance. Significantly differentially expressed genes can be defined by using an appropriate combination of number of genes to follow up and tolerable false positive rate. Descriptions for the various abbreviations: *rawp*: unadjusted p-values, *Bonferroni*: Bonferroni adjusted p-values (strong control of the FWER (Family Wise Error Rate, the probability of at least one false positive)), *SidakSS*: Sidak's single step method

6.4 Correcting the p values for multiple testing

adjusted p-values (strong control of the FWER), *SidakSD*: Sidak's step down method adjusted p-values (strong control of the FWER), *Holm*: p-values adjusted using the method from Holm (strong control of the FWER), *Hochberg*: p-values adjusted using the Hochberg method (strong control of the FWER), *BH*: p-values adjusted using the method proposed by Benjamini and Hochberg (strong control of the FDR (False Discovery Rate, expected proportion of false positives among the rejected hypothesis)), *BY*: p-values adjusted using the method from Benjamini and Yekutieli (strong control of the FDR).

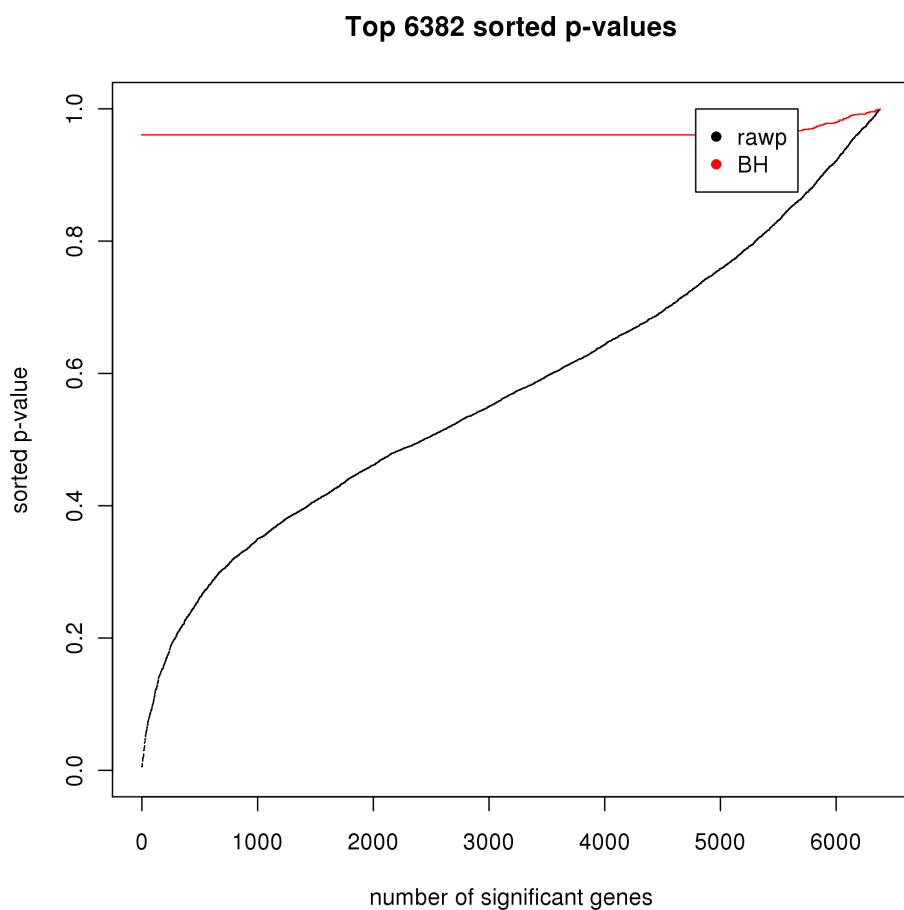


Figure 6.3: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

6.4 Correcting the p values for multipletesting differentially expressed genes using test statistics

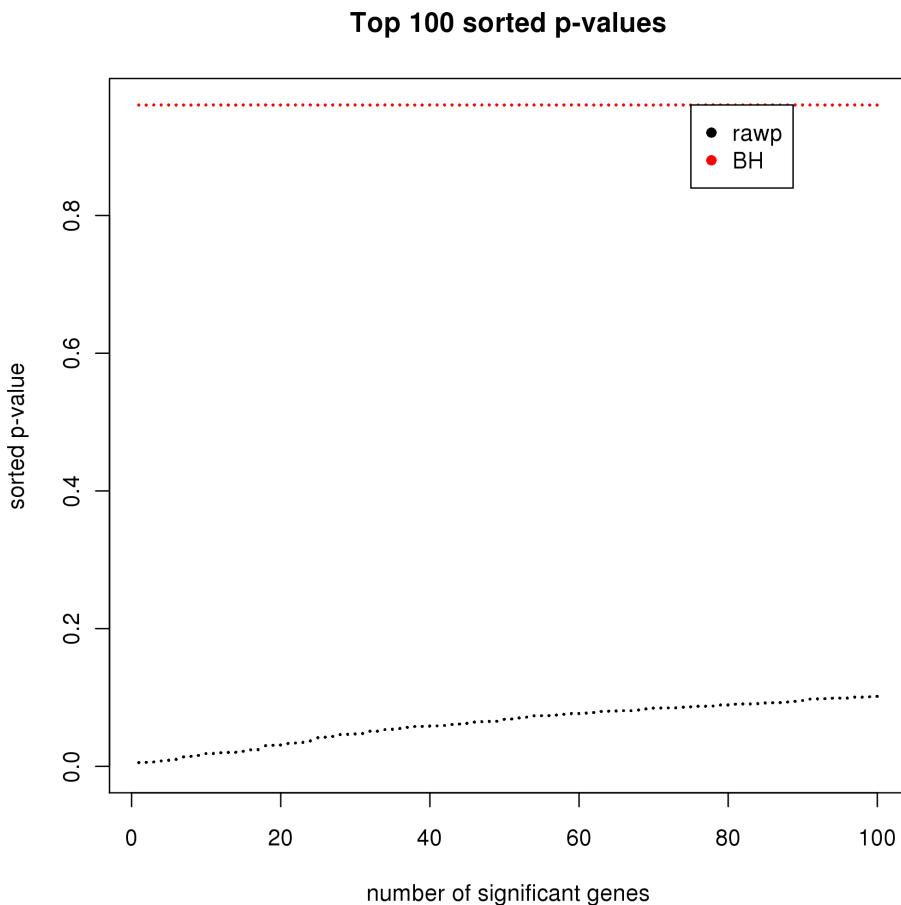


Figure 6.4: Plot of the sorted p-values. A description of the plot and the abbreviations is given in the text.

```
> Filename <- checkExistantFile("FERGUS.txt")
```

A table containing all calculated p values (raw p values and corrected p values) is saved as tabulator delimited text file to the file: *FERGUS.txt*; Calculating average regulation (M) and average expression (A) values between the two groups. The average M value for each gene is calculated by subtracting the average expression value of the gene in group 0 from the average expression value of the gene in group 1 (so a mean M of 1 means a two fold increase in the expression level of the gene in group 1 compared to the expression level in group 0).

The average MA plot is drawn using M and A values that are calculated from the average expression values of each gene in each sample group. To calculate the average ,the mean function is used.

```
> library(geneplotter)
> MAColor <- densCols(M, A, colramp = colorRampPalette(rev(brewer.pal(9,
+ "Blues"))[2:9]))
```

6.4 Correcting the p values for multipletesting differentially expressed genes using test statistics

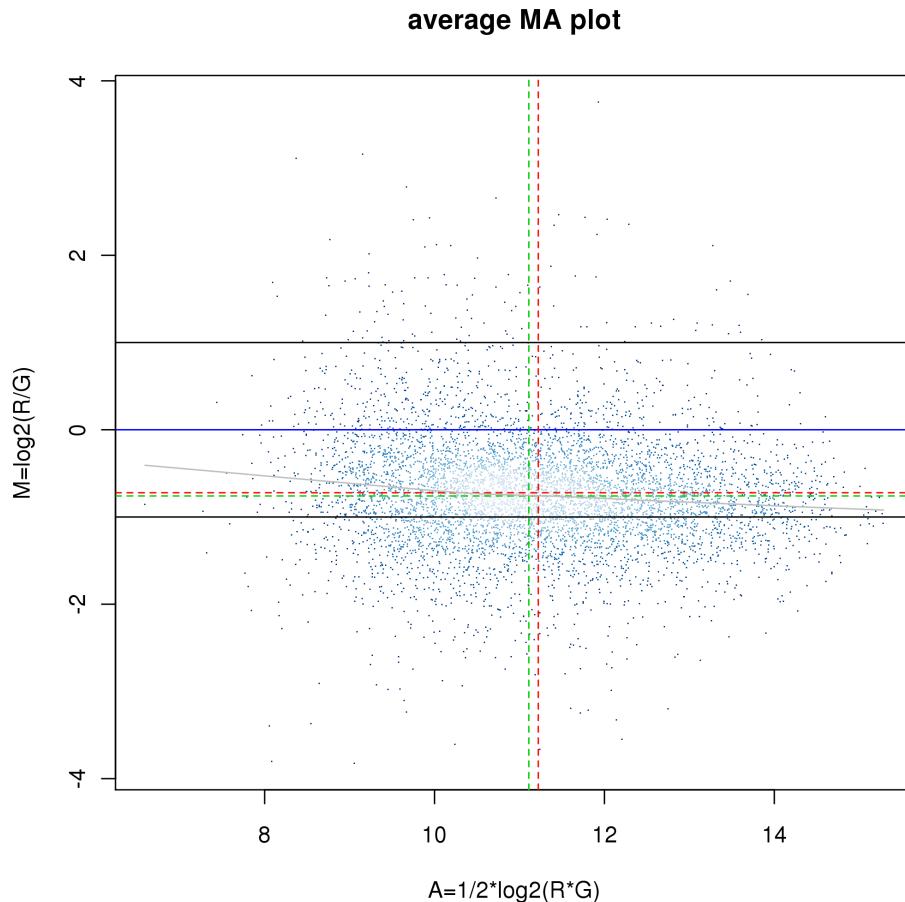


Figure 6.5: MA plot comparing the average expression values per gene from group 1 against those from group 0. Points are colored according to the local point density. White codes for high, blue for low point density.

In the volcano plot the p values are scattered against the regulation values (average M values). The volcano plot in figure 6.6 represents the average M value per gene and the according raw p value calculated using the selected test statistics. The most interesting genes would be those that have both small p values and big average M values.

6.4 Correcting the p values for multiple testing

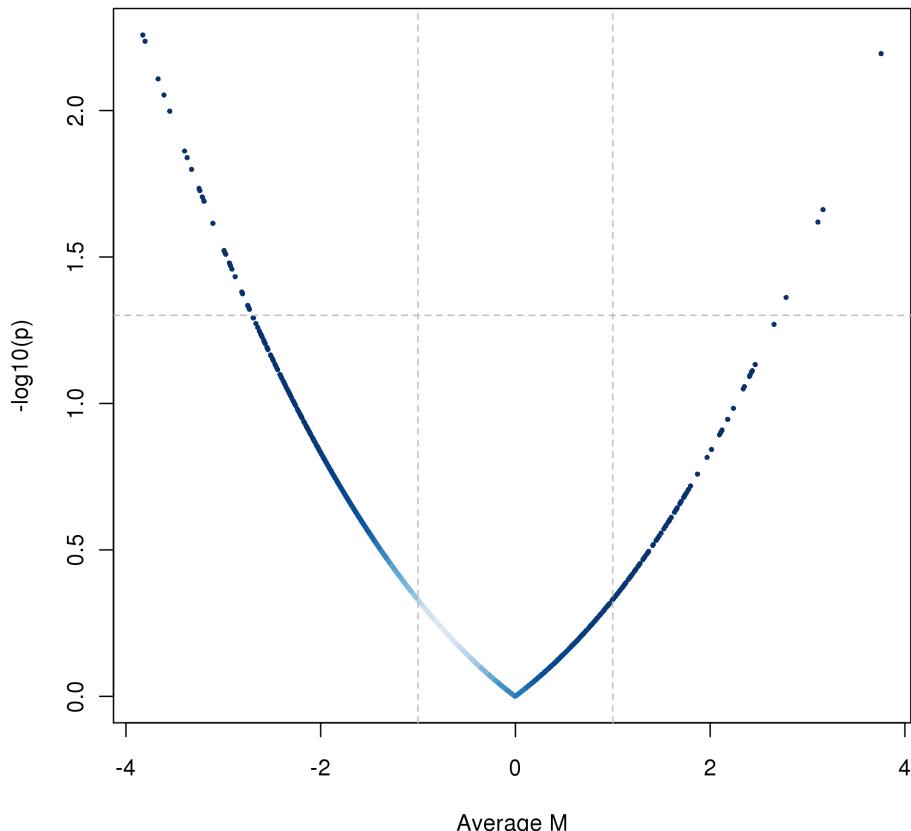


Figure 6.6: Volcano plot scattering the average M values (x axis) against the raw p values (y axis, -log10 scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

Volcano plot of the average M values against the p values corrected for multiple testing using the method proposed by Benjamini and Hochberg (figure 6.7).

6.4 Correcting the p values for multipletesting differentially expressed genes using test statistics

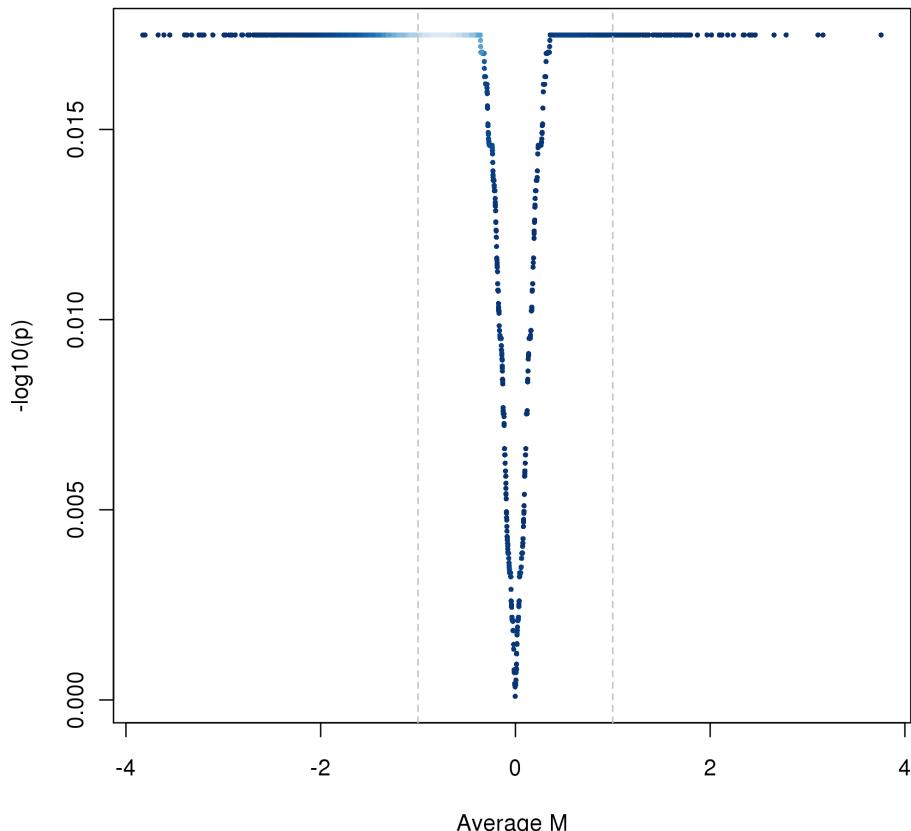


Figure 6.7: Volcano plot scattering the average M values (x axis) against the p values corrected with Benjamini and Hochbergs method (y axis, $-\log_{10}$ scale, small p values have big y values). Points are colored according to the local point density. White codes for high, blue for low point density.

The p values together with the data on which the test statistic was calculated of the 100 genes with the smallest p values is saved to the file : *FERGUS-top100.txt*